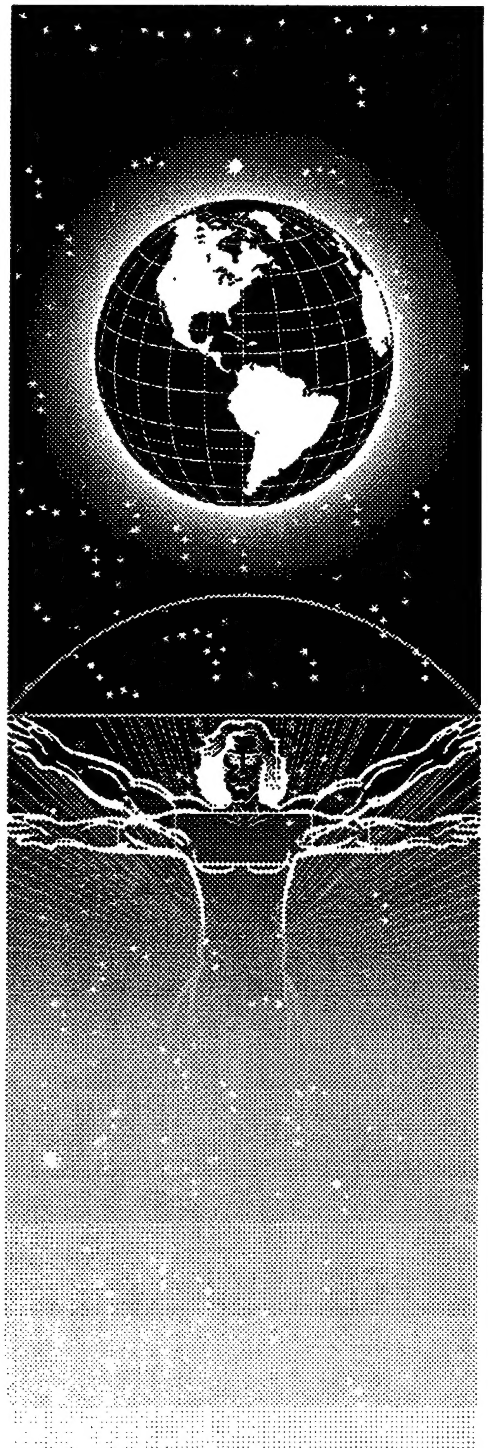
AL/CF-TR-1997-0149



UNITED STATES AIR FORCE
ARMSTRONG LABORATORY

Computer Aided Systems Human Engineering: Performance Visualization System A Hypermedia Tool for Designers

Janet E. Lincoln

HUDSON RESEARCH ASSOCIATES
41 FERRY ROAD
STUYVESANT NY 12173

Donald L. Monk

CREW SYSTEMS DIRECTORATE
HUMAN ENGINEERING DIVISION
WRIGHT-PATTERSON AFB OH 45433-7022

AUGUST 1997

INTERIM REPORT FOR THE PERIOD JANUARY 1990 TO NOVEMBER 1996

Crew Systems Directorate
Human Engineering Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

NOTICES

TECHNICAL REVIEW AND APPROVAL

AL/CF-TR-1997-0149

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

JOHN F. KENT, COL, USAF, BSC
Acting Chief
Human Engineering Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1997

3. REPORT TYPE AND DATES COVERED
Interim, January 1990-November 1996

4. TITLE AND SUBTITLE
Computer Aided Systems Human Engineering: Performance Visualization System--A Hypermedia Tool for Designers

5. FUNDING NUMBERS
C: 41624-94-D-6000
PE: 62202F
PR: 7184

6. AUTHOR(S)
Janet E. Lincoln
Donald L. Monk

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Hudson Research Associates
41 Ferry Road
Stuyvesant NY 12173

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory, Crew Systems Directorate
Human Engineering Division
Human Systems Center
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL/CF-TR-1997-0149

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The Computer Aided Systems Human Engineering: Performance Visualization Subsystem (CASHE:PVS) is a multidocument ergonomics database on CD-ROM. The reference documents in the database provide data, phenomenon descriptions, models, and standards from over 70 research areas dealing with the perceptual and performance capabilities of the human operator. This information is integrated into an interactive multimedia system that combines state-of-the-art information retrieval, browsing, and navigation with specialized tools that help the designer interpret and apply the ergonomics data available in the product. Behavioral data and phenomena descriptions in text, figures, and tables are accompanied by prototyping simulations, animations, and audio demonstrations that allow users to experience important perceptual and performance phenomena for themselves and provide a rich understanding of how these phenomena relate to the design of human-operated systems. This report provides a system overview of the software and gives detailed discussions of various design and implementation issues associated with the production of the software and its databases.

14. SUBJECT TERMS
Human factors, Human Perception, Visualization, Hypermedia, Design

15. NUMBER OF PAGES
208

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

This page is intentionally left blank.

PREFACE

This report documents the work performed to develop the Computer Aided Systems Human Engineering: Performance Visualization System (CASHE:PVS) Version 1.0 software. CASHE:PVS is part of a continuing effort of the Human Engineering Division at Armstrong Laboratory to provide usable ergonomic information to the design engineering community. The genesis of CASHE:PVS dates back to the Integrated Perceptual Information for Designers (IPID) program. Although early IPID products were issued in conventional paper form, Dr. Kenneth R. Boff and his colleagues recognized from the beginning that having electronic versions of the human perceptual and performance data would be a tremendous advantage for human-system designers. They saw that computer support systems could significantly enhance the communication of scientific data, especially when the implications for design were integrated with the data. CASHE:PVS is an implementation of that pioneering vision.

The CASHE program was accomplished under USAF project 7184, task 12, work units 21 and 22, and task 26, work unit 12. It was managed through the office of the Design Technology Branch of the Fitts Human Engineering Division. Dr. Kenneth R. Boff was the founding project manager. After he became Division Chief of the Fitts Human Engineering Division in 1991, Mr. Donald L. Monk was program manager. Fulfilling a multitude of assignments, Dr. Janet E. Lincoln, of Hudson Research Associates, was the primary consultant for CASHE, from its earliest beginnings until its publication.

It is a difficult task in all projects to appropriately acknowledge the many individuals who contributed to its success. This task is further complicated by the many and varied roles assumed by so many people. All of these individuals deserve thanks — far more than can be given here — for a job so well done. To any individuals who made contributions that have been inadvertently omitted, we sincerely apologize.

Sponsoring Agencies

- Armstrong Laboratory; Wright-Patterson AFB, OH
  Charles Bates, Jr. | Kenneth Boff | Jim Brinkley

- Air Force Office of Scientific Research; Washington, DC
  Al Fregly

- Army Research Laboratory (ARL); Aberdeen Proving Grounds, MD
  Mark Hoffmann | Dean Westerman

- Office of Naval Research; Washington, DC
  Stan Collyer

- Navy Command Control and Ocean Surveillance Center (NCCOSC); San Diego, CA
  Jeff Grossman

- Defense Technical Information Center (DTIC); Alexandria, VA
  Forrest Frank

- Federal Aviation Administration (FAA); Washington, DC
  Clay Foushee | Phyllis Kayten | Ronald Lofaro
  Ron Simons | Bill White

- NATO's Advisory Group for Aerospace Research and Development (AGARD), Aerospace Medical Panel (AMP); Paris, France
  Bill Lyle | Rob Poisson | John Winship

Developing Organizations

- Armstrong Laboratory, Human Engineering Div.; Wright-Patterson AFB, OH
  Mark Cannon | Dave Curry | Tanya Ellifritt
  Lew Hann | Ralph Korthouer | Nick Longinow
  Robert Osgood | Herschel Self | Alan Straub
  Brian Tsou | Maris Vikmanis

- Armstrong Laboratory, Biodynamics Div.; Wright-Patterson AFB, OH
  Kathy McCloskey | Richard McKinley | Charles Nixon

- CD Solutions; Dayton, OH
  Jerry Warner

- Data Conversion Laboratory; Fresh Meadows, NY
  Richard Barth | Mark Gross | Michael Gross
  David Silverman

- Gae Xavier Studios; Austin, TX
  Mike Chen | Donna Elkins | Elinor Gamache
  Yang Si | Keith Snodgrass | Tom Uthoff
  Caesar Xavier | Gae Xavier

- Logicon Technical Services, Inc. (LTSI); Dayton, OH

Keith Adams	Rebecca Donovan Brown	Chris Calhoun
Tom Cona	Eleanore Chuang	Scott Grigsby
Ann Hartung	Larry Hettinger	Glenn Johnson
Bill Janson	Roy Livingston	Susan Pfaadt
Brian Porter	Walt Rosenthal	David Spangler
Sarah Swierenga	Maxwell Wells	Lawrence Wolpert

- Institute for Sound and Vibration Research, University of Southampton; Southampton, England

  Michael Griffin Chris Lewis

- NewMedia, Inc.; Columbus, OH

Fritz Anderson	Dave Brownstein	Gary Kacmarcik
Marty Pochmara		

- Ohio State University; Columbus, OH

  Richard Jagacinski

- Search Technology; Atlanta, GA

John Baldwin	Bill Cody	Robert Glushko
Charles Howard	David Resnick	Bill Rouse
Daryl Savell	Dan Sewell	Mark Weaver
Brad Wiederholt		

- University of Dayton; Dayton, OH

  William Moroney Leslie Whitaker

- University of Dayton Research Institute (UDRI); Dayton, OH

Frank Beitel	Vicky Chambers	Don Dreesbach
Gina King	Jeff Landis	Larue Meehan
Ruth Rodak	Dennis Stafford	Tom Straw

- Wright State University; Dayton, OH

  John Flach

We would also like to thank Rebecca Donovan Brown and Glenn Johnson for assisting in the preparation of this report.

Janet E. Lincoln
Stuyvestant, NY

Donald L. Monk
Wright-Patterson AFB, OH

Trademark Notices

Apple, the Apple logo, and Macintosh are registered trademarks of Apple Computer, Inc. PowerBook, PowerBook Duo, Finder Thread Manager, Balloon Help, MoviePlayer, and QuickTime are trademarks of Apple Computer, Inc.

Author/Editor is a trademark of SoftQuad, Inc.

Canvas is a trademark of Deneba Systems, Inc.

Expressionist is a registered trademark of Prescience Corp.

Macromedia is a registered trademark of Macromedia, Inc. Director is a trademark of Macromedia, Inc.

Microsoft is a registered trademark of Microsoft Corporation. Word is a trademark of Microsoft Corporation.

SoundEdit is a trademark of Farallon Computing.

SuperCard is a registered trademark of Allegiant Technologies, Inc.

TABLE OF CONTENTS

Page

List of Figures x

List of Tables xii

1. Introduction **1**

1.1 Background 1

1.2 System Overview 3

1.3 Design Issues 8

2. Design of the User Interface **10**

2.1 Functional Objectives of the User Interface 10

2.2 General Structure of the Database 11

2.3 Information Presentation Format 13

3. Information Presentation in CASHE: The Component Viewers **17**

3.1 TextViewer 20

3.2 TableViewer 23

3.3 FigureViewer 27

3.4 Entry Palette 32

4. Information Access in the Reference Database **35**

4.1 Browsable Outlines 35

4.2 Text Search 41

4.3 Hyperlinks 47

4.4 Windows Menu 50

4.5 History Menu 50

4.6 Bookmarks List 51

Page

5. User Customization **52**

5.1 Annotations 52

5.2 Session Support 60

5.3 Export and Import of Information 60

6. Special User Aids **63**

6.1 Glossary 63

6.2 Design Checklist 64

6.3 DataDigitizer 65

6.4 On-Line Help 68

7. Visualization Enhancements **70**

7.1 Play-Only Demonstrations 71

7.2 Perception and Performance Prototyper Test Benches 73

8. Data Preparation **93**

8.1 Text 95

8.2 Figures 100

8.3 Tables 128

8.4 Other Data Files 134

8.5 Indexing Files 137

9. System Software **143**

9.1 Information Viewers 144

9.2 Entry Palette 160

9.3 Retrieval Module 162

9.4 CASHE Server 165

9.5 Session Manager 166

9.6 Residual Engine Functions 168

9.7 Software Testing 170

Page

10. Integrated External Applications **171**

10.1 Test Benches 171

10.2 On-line Help 176

10.3 DataDigitizer 177

11. Product Release **178**

11.1 Packaging 178

11.2 Installation 180

11.3 Customer Support 181

12. Future Directions **182**

12.1 Scale-Up 182

12.2 Future Enhancements 184

13. CASHE Bibliography **188**

Appendix A **190**

LIST OF FIGURES

Figure *Page*

1. Structure of the database 15
2. Standard interface elements for component viewers 18
3. Sample TextViewer window 21
4. Sample TableViewer window 24
5. Sample FigureViewer window 29
6. Entry Palette 33
7. Sample OutlineViewer window 40
8. Query dialog box 43
9. Find dialog box 46
10. Examples of text hyperlinks 49
11. Dialog box for creating a bookmark 53
12. Session Bookmarks window 54
13. Dialog box for creating a user note 56
14. Session Notes window 57
15. Link destination dialog box 58
16. Link palette for creating a new hyperlink 58
17. Session Link Sources window 59
18. Sample questions from the Design Checklist 65
19. Sample DataDigitizer window with open Data Table window 67
20. Example of Balloon Help 69
21. Sample screens from play-only demonstrations 72
22. Standard TextBenchViewer window 82
23. Sample topic selection menu 84
24. Main screen from a typical standard topic module 86
25. Typical Settings window display 87
26. Typical Results dialog box display 89
27. Typical Results window display 90
28. Typical control panel for Custom Options topic 91
29. Structure of figure components in the database 102
30. Application of split-panels method 105

Figure		Page
31.	Application of merged panels method	106
32.	Application of data overlay method	107
33.	Application of explanatory overlay method	109
34.	Application of linked panels method	111
35.	Application of preview-and-zoom method	113
36.	Application of clipped view method	114
37.	Graphic displayed on-line using a combination of methods	116
38.	One panel of a linked-panel table showing pop-up graphic	131
39.	Test bench development process	172
A-1.	Control flow for functions accessible from the TextViewer	191
A-2.	Control flow for functions accessible from the TableViewer	192
A-3.	Control flow for functions accessible from the FigureViewer	193
A-4.	Control flow for functions accessible from the entry palette	194
A-5.	Control flow for the Query function	195
A-6.	Control flow for viewer annotation functions	196
A-7.	Control flow for functions accessible from the DataDigitizer	197

LIST OF TABLES

Table *Page*

1. General subject areas of the 71 play-only demonstrations 71
2. P^3 test benches developed for CASHE version 1.0 78
3. Methods used to display complex or oversized figures 118
4. Query stop words 142
5. Primary specialized software modules 143
6. CASHE query language 164
7. Subject areas of potential test bench topics 173

1. INTRODUCTION

1.1 Background

Over the last decade and a half, the Human Engineering Division of the Air Force's Armstrong Laboratory has carried out a series of programs aimed at improving access to human perceptual and performance information among system designers and their counterparts who acquire, evaluate, or test those systems. The underlying assumption has been that inclusion of this information during system development enhances human performance and satisfaction with the system, while failure to consider such information increases the potential for human error and dissatisfaction.

Under the Integrated Perceptual Information for Designers (IPID) program, the Human Engineering Division produced two hard-copy resources that address this issue: the *Handbook of Perception and Human Performance*, and the comprehensive *Engineering Data Compendium (EDC)*. These documents seek to provide relevant information concerning human perception and performance capabilities and their impact on system design in a form that will put it at the fingertips of those personnel directly involved in design decisions.

The Computer Aided Systems Human Engineering (CASHE) program builds on these earlier efforts. CASHE began as part of the Designer's Associate program, which examined engineers' use of technical information in design decision making and defined functional requirements for a class of automated support services to enhance the use of ergonomics information in the design process.

Based on empirical studies and analyses as well as the lessons learned from interactions with hundreds of system designers, human factors

practitioners, and other users of ergonomics information, CASHE has developed a vision of a future design support environment that fully integrates ergonomics information into the system design process. This workspace environment incorporates integrated CRTs, small-group wall displays, auditory systems, and virtual display technologies to support multidisciplinary design teams in visualizing and experiencing the operational impact of design decisions on their proposed crew system designs, even in early conceptual phases.

The first step toward this vision is the CASHE Performance Visualization Subsystem (CASHE:PVS),[1] a multidocument ergonomics database on CD-ROM. The reference documents in the database provide data, phenomenon descriptions, models, and standards from over 70 research areas dealing with the perceptual and performance capabilities of the human operator. This information is integrated into an interactive multimedia system that combines state-of-the-art information retrieval, browsing, and navigation with specialized tools that help the designer interpret and apply the ergonomics data available in the product. Behavioral data and phenomena descriptions in text, figures, and tables are accompanied by prototyping simulations, animations, and audio demonstrations that allow users to experience important perceptual and performance phenomena for themselves and provide a rich understanding of how these phenomena relate to the design of human-operated systems.

Providing ergonomics data in an electronic form has two advantages. First, presenting the information on-line enables CASHE to capitalize on the power of desktop computing to manage large volumes of information and support dynamic aids that enhance the value of the static information. Second, designers typically work in a computerized environment and depend heavily on administrative and engineering software. Providing ergonomics

---

[1] For convenience, CASHE:PVS will be referred to as "CASHE" in the remainder of this report.

information on-line makes it easier for designers to integrate CASHE with other design tools and thus increases the likelihood that designers will incorporate ergonomics data into the design process.

It should be emphasized that version 1.0 of CASHE is the first incarnation of a system that is intended to evolve continually during the lifetime of the product. Version 1.0 is an R&D product that tests the feasibility of the general approach. It does not attempt to solve all or even most of the access problems and issues identified by the IPID and Designer's Associate programs. It starts at the most important point: the beginning. Later versions of the system will tackle a larger and larger percentage of these issues as the system develops to become more mature and robust in response to the lessons learned through observing actual use.

1.2 System Overview

1.2.1 Objectives

The CASHE ergonomics database is intended to serve as an integral reference tool for designers. The objectives of the database are:

- to provide concise, definitive information to address engineers' design problems and questions in the ergonomics domain, with a minimum of user training and effort;
- to support integration of the CASHE tool with other software in the user's work environment by providing simple and cost-effective means for the user to export information out of the database;
- to allow users to augment the database with their own personal, domain-specific, and problem-specific knowledge to support individual needs and goals;
- to provide ongoing assistance to designers in acquiring a mental model of the areas within ergonomics directly applicable to their work and interests.

1.2.2 Audience

1.2.2.1 Primary users: The primary or target users of CASHE are engineers and human factors practitioners who contribute to the design, development, testing, and evaluation of machines and systems that include a human interface. Such systems encompass virtually all equipment, facilities, consumer products, and environments that humans will interact with and use.

These CASHE users are assumed to be college educated, and some will hold advanced degrees. Due to the range of technical expertise required in system development efforts, users will come from many academic specializations. These will include several engineering disciplines (e.g., electrical, mechanical, aeronautical, civil, industrial) as well as computer sciences, behavioral sciences, and medical and physiological sciences.

Users can be characterized specifically both by the need for ergonomics information and by the lack of formal training in that discipline. End users are expected to possess only low to medium familiarity with the subject matter of our product. It will be an atypical primary user who knows the human perception and performance domain in depth along with another technical specialty. Nevertheless, we want CASHE to encourage users to address human-related issues more frequently than they do now regardless of their specialization.

CASHE target users are expected to have moderate computer literacy, defined as being familiar with the use of specific tools and applications on a limited number of computer systems or platforms of particular use.

Users are projected as having the system installed in an individual-use setting. They are expected to be running the CASHE system in short, intense sessions, separated by long stretches of involvement with other issues. This is a result of the way design projects are structured: the high-level design questions are asked most often during the initial stages of design. Thus, users

will have neither the opportunity nor the inclination to become expert in the use of the CASHE system.

1.2.2.2 Secondary users: CASHE was conceived and designed to satisfy a particular set of information needs among human-machine system developers. However, the product is expected to appeal to a wider range of audiences. In particular, CASHE will be useful to researchers, educators, and students in engineering and the behavioral sciences who investigate, teach, or seek to learn about the phenomena, concepts, data, and methods of human perception and performance.

1.2.3 Platform

CASHE version 1.0 is designed to operate on Apple Macintosh computers. The minimum configuration to support CASHE version 1.0 is a Macintosh II running System 7.0 or higher, with a 13-inch diagonal color monitor, 8 megabytes (MB) of RAM, and a CD-ROM drive. Although the minimum installation requires only 4 MB of available hard disk space, a full installation, utilizing 27 MB of disk space, will permit the system to achieve significant performance advantages. The CASHE system takes advantage of additional RAM up to the limits of system expansion. It is also able to take advantage of larger monitors up to a 21-inch diagonal screen size. CASHE will run on a 256-shade gray scale monitor; however, some test bench phenomena cannot be experienced without a color display. The PowerBook and PowerBook Duo families can be used to access the reference documents. However, to run the Perception and Performance Prototyper (P^3) test benches, an external monitor of at least 13 inches must be connected as the main monitor.

The CASHE system is designed for access by one user at a time on a single platform. As there are features that are configuration dependent and determined at installation time, access over a network is not recommended.

The Macintosh hardware families on which CASHE will run have differing performance specifications; thus, CASHE will behave somewhat

differently on each platform. Overall CASHE system performance is tuned to nominal acceptability on the entry-level platform with the assumption that a full installation has been performed (i.e., the entire set of cached files has been copied from the CD-ROM to the hard disk). Performance gains due to hardware or operating system are handled by splitting the CASHE functionality (program code) into two groups. For functions in the first group (which constitute the wide majority of instances), CASHE simply reflects any hardware performance increases as a simple, unengineered speed increase (as do the majority of available applications). For functions that are by nature timing-specific (such as are found in the test benches, for example), the implementation is designed to provide a single hardware-independent timing and performance model.

The CASHE software, including the P^3 test benches, is designed to operate in off-the-shelf mode; that is, it does not require special expansion cards or custom external equipment. Stereophonic earphones are necessary for two of the test bench demonstrations. Such earphones are inexpensive and easy to purchase, and most users probably already own a pair. For one of these demonstrations, headphones with an external volume control may be necessary to hear the effects.

1.2.4 Hypermedia Documents

CASHE, Version 1.0, contains two primary reference documents that are included in machine-readable form on the CD-ROM. The documents are:

> *Engineering Data Compendium: Human Perception and Performance* (Vols. 1-4) by K. R. Boff and J. E. Lincoln, Armstrong Aerospace Medical Research Laboratory, Wright-Patterson AFB, OH, 1988.
>
> *Military Standard 1472D Human Engineering Design Criteria for Military Systems, Equipment, and Facilities (MIL-STD-1472D)*, Notice 2, Department of Defense, Washington, D.C., 30 June 1992.

CASHE also contains a supplementary feature treated as a separate document: the Perception and Performance Prototyper (P^3).

1.2.4.1 Engineering Data Compendium: The *Engineering Data Compendium: Human Perception and Performance (EDC)* is a four-volume encyclopedic reference document developed by the Human Engineering Division of the Armstrong Laboratory and co-sponsored by the Department of Defense, NASA, and NATO AGARD. The *EDC* is composed of over 1100 entries incorporating the following types of information:

- basic human perceptual and performance data;
- summary tables integrating data from related studies;
- descriptions of human perceptual phenomena;
- models and quantitative laws;
- section introductions providing an overview of topical areas; and
- background information and tutorials on specific topics to help users understand and evaluate the material in the *EDC*.

To make pertinent information more usable, graphic modes of information are used wherever possible. The *EDC* contains more than 2000 figures and tables, including data graphs, models, schematics, demonstrations of perceptual phenomena, and diagrams of methods and techniques. Other features of the *EDC* include indicators of data reliability, caveats to data application, and the use of standardized units of measurement (Système International).

1.2.4.2 MIL-STD-1472D: *MIL-STD-1472D* is a military standard providing human engineering design criteria for military systems, equipment, and facilities. The design criteria, principles, and practices contained in this standard are intended to achieve successful integration of the human into the system and to promote effectiveness, simplicity, efficiency, reliability, and

safety in system operation, maintenance, and personnel training. More than 80 figures and tables are included.

1.2.4.3 Perception and Performance Prototyper: The Perception and Performance Prototyper (P^3) is a collection of interactive test benches that allow users to explore and experience selected behavioral phenomena contained in the *EDC* and *MIL-STD-1472D*. CASHE Version 1.0 includes the following test benches.

1. Auditory Sensitivity
2. Display Vibration
3. Flicker Sensitivity
4. Manual Control
5. Motion Perception
6. Sound Localization
7. Speech Intelligibility in Noise
8. Visual Acuity
9. Visual Optics
10. Visual Search
11. Warnings and Alerts

These test benches, as a group, comprise P^3.

1.3 Design Issues

In assembling the CASHE reference database and developing the interface for it, the goal was to improve the accessibility and interpretability of the information by implementing state-of-the-art hypertext/hypermedia techniques while remaining faithful to the intent of the original documents. To achieve this goal, a range of issues had to be explored concerning:

- how to convert documents to hypertext and hypermedia;
- how to portray and integrate multiple documents;
- how to represent complex graphics, tables, and illustrations on small desktop displays;
- how to support information access and navigation in large-scale hypermedia;
- how to help users ask "correct" questions of the system;
- how to aid users in understanding and interpreting the technical information in the documents;
- how to design an architecture that supports future modifications and integration of new information.

The purpose of this report is to document some of the solutions we found and lessons we learned in dealing with these issues. We believe our experiences can be helpful to others embarking on similar technical reference projects. The remainder of this report details the approaches we used and methods we employed in converting printed documents to electronic form, creating an interface to provide quick and easy access to the material, and developing interactive interpretive tools for users.

Chapters 2-7 discuss the CASHE user interface: how users interact with the product, what functions they can perform, how database information is presented to them, and what special aids are available to help them understand and interpret this information. Chapters 8-10 outline the system architecture in very general terms, including the data structures and software modules that support the interface functionality. Chapter 11 describes consumer packaging. We close in Chapter 12 by briefly enumerating some possible enhancements for later versions of the CASHE database and outlining future directions for the product.

2. DESIGN OF THE USER INTERFACE

2.1 Functional Objectives of the User Interface

In designing the CASHE user interface, our aim was to make it easy for users to interact with the product and minimize the time spent learning to use it. Once users are familiar with the operation of CASHE, they should be able to perform the necessary functions rapidly and efficiently, so that their effort and energy in interacting with the database can be spent on analyzing and evaluating the information they have recovered, not on trying to get the software to do what they need it to do.

Keeping in mind that the overall goal of the CASHE database is to improve the accessibility and usability of ergonomics data in a system design environment, we defined a set of functions the interface had to support in order to achieve this goal. In particular, when running the CASHE software, the user must be able to:

- access the CASHE documents in a variety of ways and from a number of different perspectives, so that it will be easy to find pertinent data that answer the user's design questions;
- view different types of data (such as text, tables, and figures) in various combinations to explore and compare different aspects of behavioral information;
- directly experience some of the perceptual and performance phenomena described in the CASHE documents;
- perform simple and complex search queries on the contents of the documents to find specific information;
- attach notes to information of particular interest;

- give alternate names to locations in the information space and mark them for easy recall;
- print some of the forms of information from the system in a usable hard-copy form;
- establish relationships (links) among information items so that linked items can be retrieved rapidly;
- export some forms of the information from the system to other software packages;
- return to recently viewed data for review or reorientation;
- save specific configurations of database information and annotations, so a previous work context can be reinstated.

The next few chapters of this report describe the details of the CASHE interface design that implemented these functional objectives.

In developing the user interface for the CASHE database, we also strived to maintain consistency with established standards. Apple Computer has defined a general philosophy for graphical interface design for the Macintosh and has published guidelines for software developers that embody this philosophy and create a uniform look and feel for all Macintosh applications. We adhered to this Macintosh model whenever possible to provide a compatible product that could capitalize on users' familiarity with other Macintosh applications.

2.2 General Structure of the Database

To enable users to interact easily with the CASHE database, we considered it essential to provide a uniform interface to all the reference sources on the CD-ROM. This raised the serious pragmatic and philosophical question of whether to port documents intact into the product or augment their presentation to take full advantage of hypermedia capabilities. Re-authoring documents is a costly and highly uncertain process. On the other hand, from

an ergonomic standpoint, failure to capitalize on the advantages of hypermedia means lost opportunities to improve the usability of the product. Because version 1.0 of CASHE is viewed as an R&D product, we decided to experiment with recasting all documents to conform to a general structure that would permit the use of advanced techniques for data retrieval, browsing, navigation, and display.

The granular level at which reference sources in CASHE Version 1.0 are accessible to the user is the "entry." An entry is a self-contained unit of information dealing with a fairly narrow, well-defined topic. Each electronic entry is addressable by key terms and cross-referencing and is the destination level for access structures. Each entry has a number and a title identifying the topic covered in that information unit.

The *Engineering Data Compendium (EDC)*, one of the two reference documents on the CASHE CD-ROM, was originally designed with the goal of presenting ergonomic information in a standardized, easy-to-use format. It was already parsed into consistent entry units, most no larger than two printed pages, with a unique entry number, descriptive title, and highly uniform substructure. This organization made it very amenable to conversion to hypertext. *MIL-STD-1472D* is a hierarchically organized document consisting of numbered sections and subsections. Unlike the *EDC*, however, sections in *MIL-STD-1472D* have no consistent substructure and are not equally detailed. For example, the lowest-level subsection in a given major section may be numbered with anywhere from two to six digits and is generally a single paragraph of no more than 6-8 lines. To make *MIL-STD-1472D* easier to use in the CASHE product, an entry structure was overlaid on it by grouping the original small subsections into 41 larger entry units that seemed naturally related by their content. The original paragraph-numbering sequence of the document is strictly maintained. The number for an entry is derived from the original document—the paragraph number of the first paragraph in the entry is the entry number. Entry titles are generally taken from the initial paragraph title, but sometimes have been slightly modified to

be more descriptive of the contents of an entry. The new *MIL-STD-1472D* entries are approximately the same size as the *EDC* entries.

2.3 Information Presentation Format

All of the data contained in the hard-copy editions of the reference sources are included in the electronic version. The physical layout differs from the printed documents, however. A special electronic format was developed that would meet the information requirements of designers within the limited physical screen size and support the hypermedia functionality desired for the interface.

From its beginnings in the IPID program which preceded CASHE, one of the primary goals of the project has been to develop ways of presenting human perceptual and performance information that enhance the usability of this information for system designers. A major contribution of the *EDC* was the introduction of a special format for presenting complex behavioral data that made it easier for individuals with little or no background in the subject matter to understand and use this information.

A major challenge in designing the presentation format for the CASHE CD-ROM was to meet, in the electronic version, the same information communication objectives that guided the development of the printed version of the *EDC*. We re-framed these objectives for the new electronic medium as follows:

1. The topic treated in each entry should be easily and rapidly identifiable. Users should be able to tell at a glance what a given entry is about so they can judge its relevance to their current needs.

2. The format should allow a quick overview of the structure and content of each entry. Just as readers of a book might flip the pages of a chapter to see what figures and tables are provided and what text sections there are, users of the CD-ROM should be able to get a sense of the information content of an entry before they begin reading it.

3. **The design should help users maintain context, both within an individual entry and in CASHE as a whole.** One of the perils of hypertext documents is that users can easily become disoriented and lose their sense of where they are in the document.

4. **Figures and tables should be given prominence.** Engineers are used to dealing with quantitative data and prefer that information be presented in the form of graphs or tables.

5. **Information should be easy to locate within individual entries.**

6. **The format should support users with different goals, needs, backgrounds, and levels of knowledge.**

7. **Reader aids and supplementary material should be readily accessible and easy to invoke.** Users should not continually have to perform the electronic equivalent of "flipping to the back of the book" to utilize access routes and call up background material such as glossary definitions.

8. **The design should make it easy for users to compare data from different entries and extract material for their personal use.** Users of the paper documents can pull out individual pages to compare data from different locations or to photocopy entries of special interest. We need to provide the electronic equivalent of these functions.

To implement these objectives in an electronic medium, we had to conceptualize the material somewhat differently than it is generally viewed on paper. One problem with the current generation of hypertext and hypermedia systems is that a page-oriented presentation of information is often retained as an artifact of printed documentation. Users are therefore constrained to an inflexible page layout of information and may suffer significant inconvenience in trying to compare or relate information on different "pages."

To avoid this problem, CASHE adopts a different approach to the database information. CASHE separates the information in the database

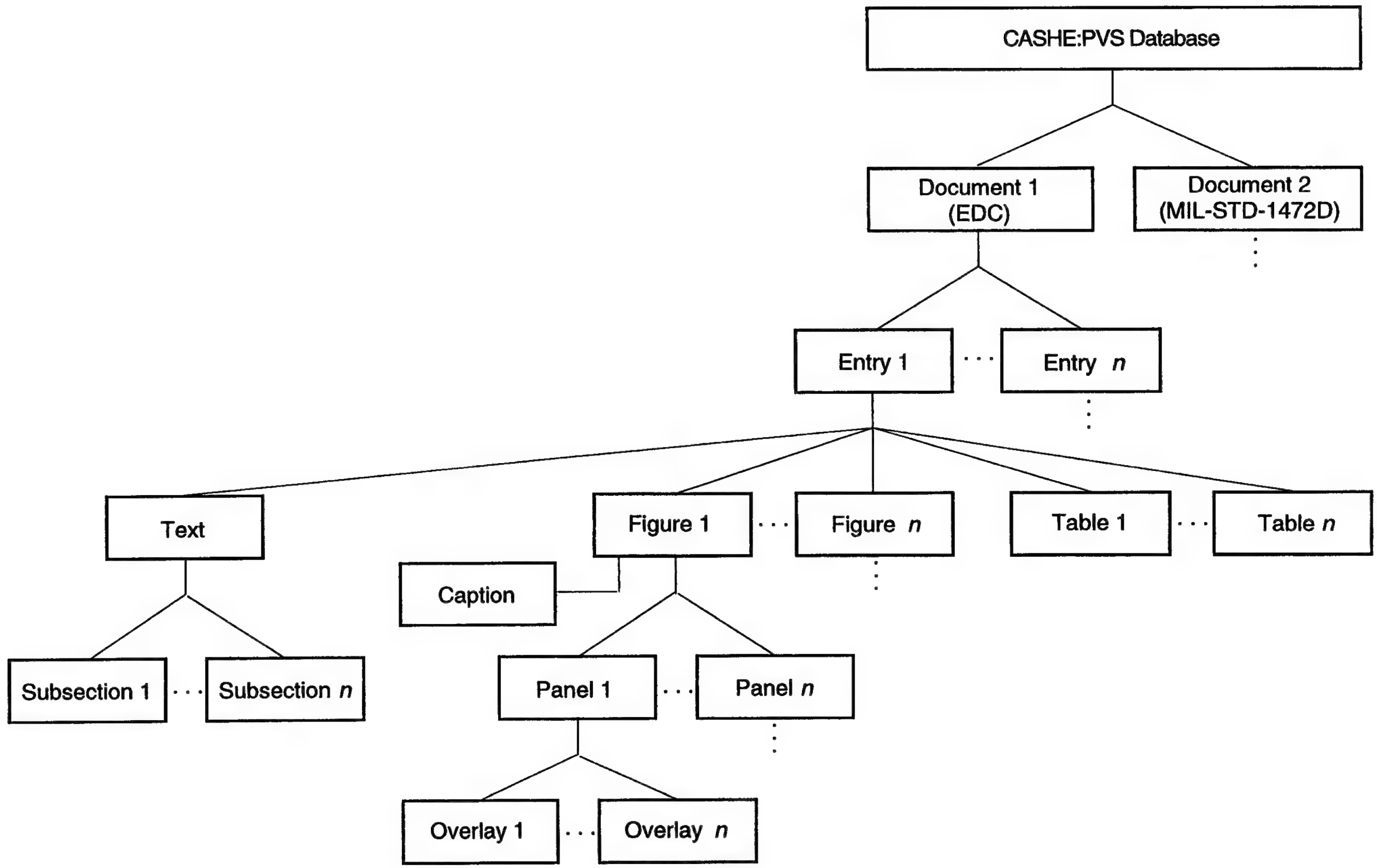


Figure 1. Structure of the database

documents according to fundamental data types and uses a specialized, dedicated viewer for each data type. Each electronic entry in the reference documents is subdivided into three data components—text, table, and figure. These components may be further broken down into panels (figures and tables) and subsections (text). Figure 1 shows the structure of the database. Every entry has a single text component. An entry may also have one or more figure components and one or more table components.

Each component is displayed in a specialized viewer—TextViewer, TableViewer, or FigureViewer—that provides the user easy access to the information in a logical and coherent framework. There are two advantages to separating the data types. First, because each viewer operates on a specific kind of data, it can provide unique features and functionality tailored to the class of information being displayed. Second, separating data types into different windows makes it easy for users to view different forms of essentially the same information side by side or to compare data from different entries. For example, users can place a window containing a discursive description of experimental results next to a window showing a graphic plot of the data collected or a table outlining the experimental stimuli and procedures. Or, a data graph from one entry plotting auditory sensitivity in quiet can be opened next to a graph from another entry showing auditory sensitivity in noise.

The next chapter examines the three CASHE component viewers and describes the features and functions of each in more detail.

3. INFORMATION PRESENTATION IN CASHE: THE COMPONENT VIEWERS

As described in the previous chapter, the information in the CASHE database is separated by data type into text, figure, and table components. Each component class is displayed separately in a customized viewer—TextViewer, FigureViewer, or TableViewer. Each component viewer is a special window type that understands the intrinsic properties of the data class displayed in that viewer and provides only those operations that are compatible with this data class. Although each viewer has unique features, all three viewers provide a core set of information-handling functions and presentation attributes.

The three component viewers are designed to have a standard interface, so that features or elements common to more than one viewer always look the same. Since other applications may be running at the same time as CASHE, there may be non-CASHE windows on the desktop along with the component viewers. A consistent and distinctive appearance for CASHE component viewers helps users maintain context.

Figure 2 shows the standard interface elements for all component viewers. The *title bar* of the viewer window identifies the document, entry number, and component (text, figure number, or table number) to preserve a sense of place within the database. In addition, an *identification region* at the top of the viewport area displays the entry number and full entry title. For the *EDC*, the topic area and subarea in which the entry is located are also displayed (this hierarchical information is similar to running heads in paper documents). This on-screen format is designed to minimize the disorientation that is a common problem in hypertext databases.

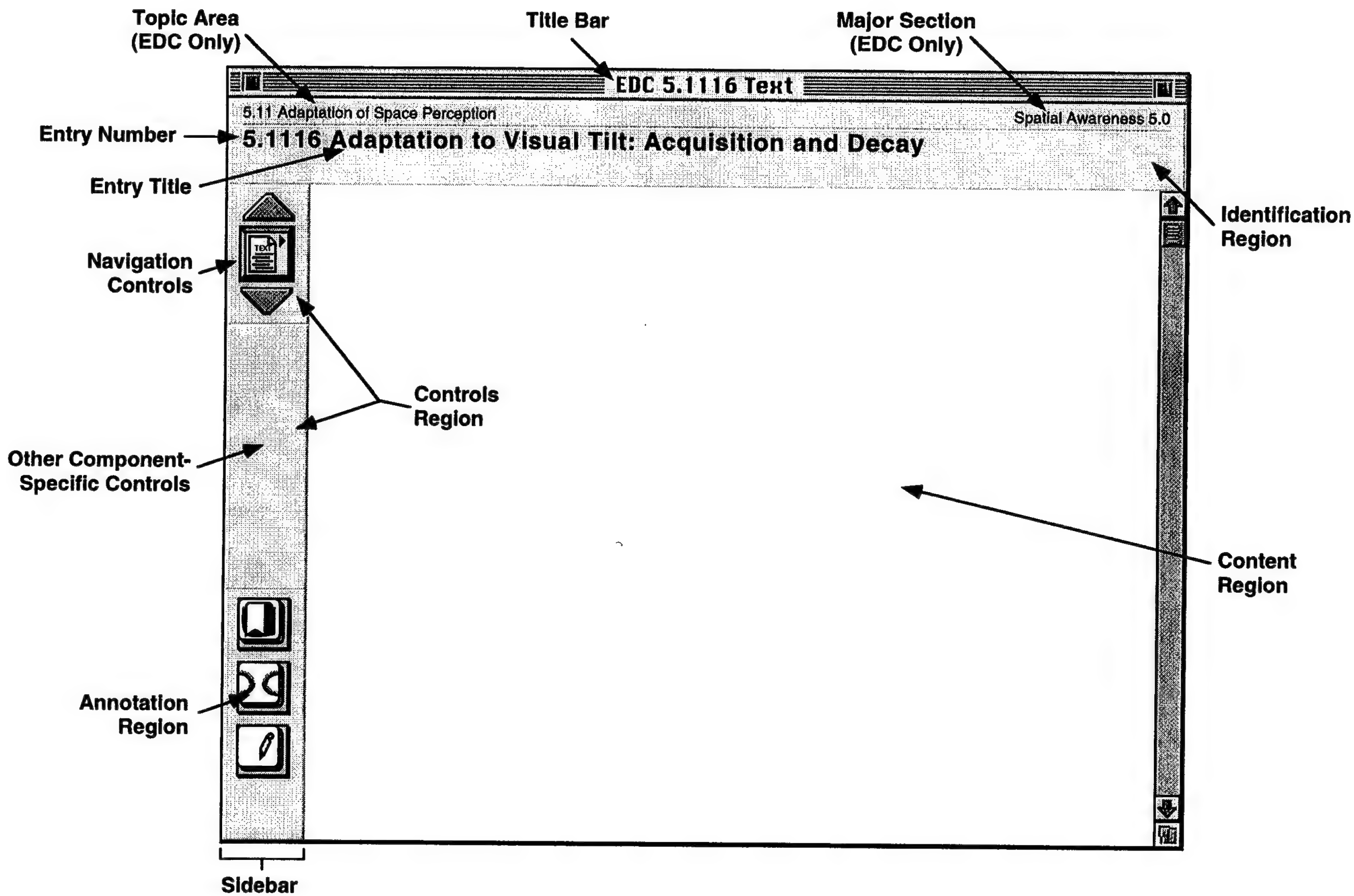


Figure 2. Standard interface elements for component viewers

Navigation controls in each viewer let users move easily among all components of the given information type in the entry. A component selector pops up a menu of all the figures or tables available for the entry (or all the text subsections for text components). Selecting an item allows the user to jump directly to that component or text subsection. Arrow controls let users browse from subsection to subsection in the text or from one table or figure to another.

The *annotation region* provides controls for creating and accessing user-defined information associated with the component of the entry that is currently displayed in the viewer. Users may attach notes to the information, create hyperlinks from the current component to other related components, or place a bookmark to aid in returning to the current component. (Annotation functions are discussed in greater detail in section 5.1.)

The *content region* displays the content of the given text, figure, or table component of the entry. It may contain embedded buttons and hot spots specific to the particular component in view to aid users in navigating to other portions of the component or to other relevant components. The appearance of the text in the content region (such as the font and style for titles and subheadings, body text, table body and column headings, and footnotes) is determined by an internal style sheet for each component type.

Users can open as many different TextViewer, FigureViewer, and TableViewer windows as desired, in any mix, up to the limits of computer memory. Viewer windows can be scrolled, resized, and repositioned on the desktop to permit easy comparison of information. To preserve context and access to viewer functions, the identification region (entry number and title) and the sidebar (buttons and controls) do not scroll with the window contents but remain fixed in view. In addition, the entry title wraps when the window is resized horizontally, so the title remains readable.

Viewer window contents (including material scrolled out of view) can be printed or exported in a form that is readable by other applications. Internal style sheets for printing and export control the presentation attributes of each

component type. Since these style sheets are separate from those for on-screen presentation, the appearance of a printed or saved component may be different from its appearance on the computer monitor.

3.1 TextViewer

The TextViewer is the component viewer for text. Each entry has a single text component, which is further subdivided into multiple text subsections. For example, the text components of most *EDC* entries have eight standard text subsections with titles like Key Terms, General Description, and Experimental Results. The text components of *MIL-STD-1472D* entries have text subsections consisting of numbered titles and paragraphs.

Figure 3 shows a sample TextViewer window. When utilizing a TextViewer, the user may:

- Select and jump to any subsection within the text;
- Resize the window, which causes the text to be reflowed into the content region;
- Scroll the text vertically using the standard Macintosh scroll bar features;
- Navigate using hyperlinks embedded within the text;
- Access annotation features;
- Print the text component;
- Copy the text or export it to a file;
- Search the text in the viewer window for any character string.

Users can easily navigate among the text subsections by means of the subsection selector and arrow controls at the top of the left sidebar. Clicking the Text Subsection Selector opens a pop-up menu that lists all the subsections of the current entry's text. Selecting a subsection will scroll the TextViewer window to the desired subsection. Clicking the Previous Text Subsection (upward-facing) arrow scrolls the TextViewer to the beginning of the previous

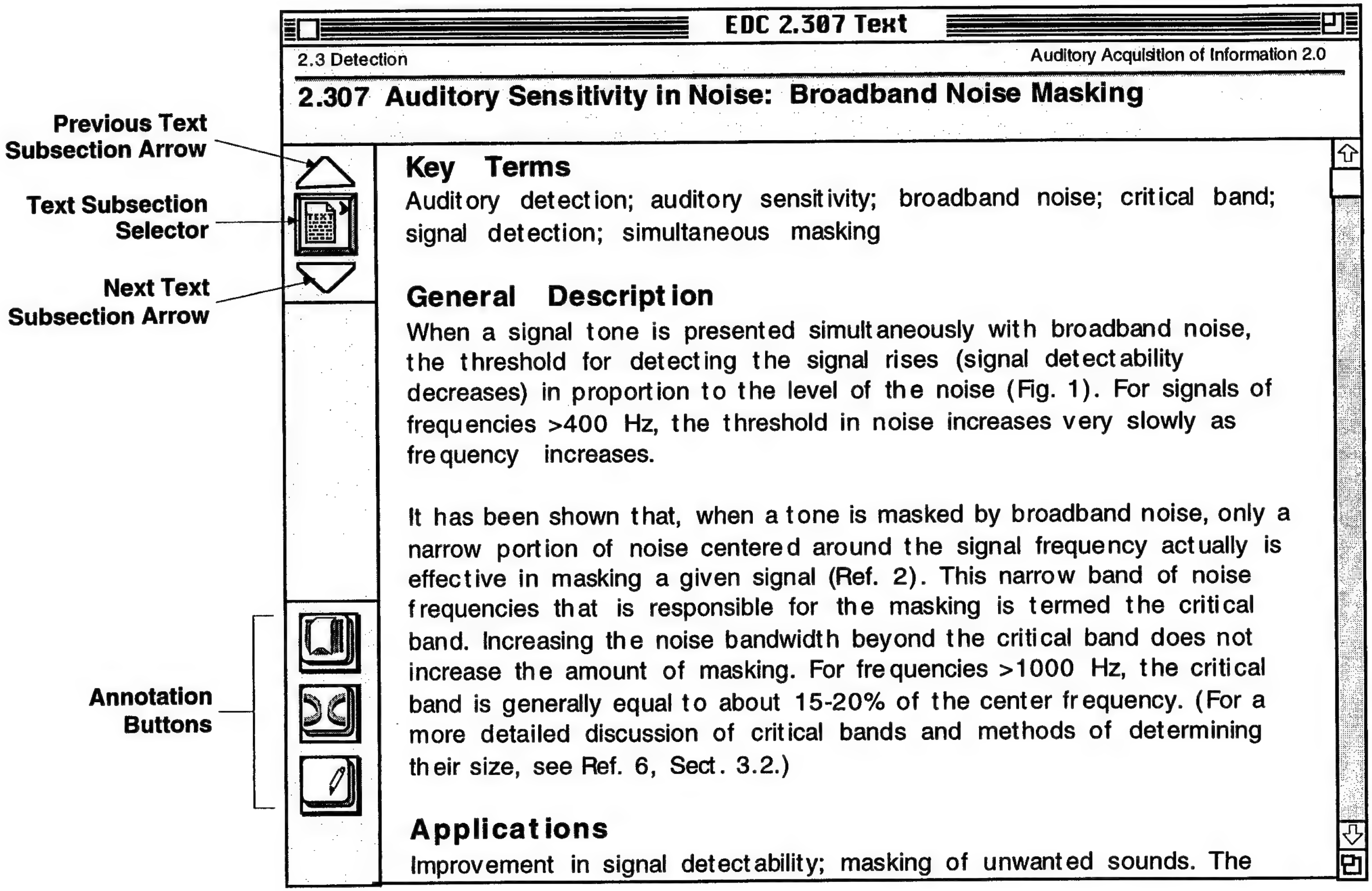


Figure 3. Sample TextViewer window

subsection. Clicking the Next Text Subsection (downward-facing) arrow scrolls the TextViewer to the beginning of the next subsection.

Using the annotation buttons on the lower left sidebar, users can set an electronic bookmark to make it easy to return to the text component displayed in the TextViewer window; create hyperlinks between the current component and another database component; or compose and attach a note to the component.

TextViewer windows are moveable and resizable. When the window is resized, the text is re-flowed to fill the content region (i.e., text never moves out of view on the right side of the viewer window). Text may be scrolled vertically to reveal additional text above or below the viewing region.

Users can select text within the TextViewer window by clicking the cursor over the beginning of the text and dragging across to the desired end point. The highlighted text may then be copied to the Clipboard for export to user notes or other applications. Text is saved to the Clipboard in rich text format (RTF), which preserves certain formatting attributes such as font family, boldfacing, and italic.

All of the text in an active TextViewer window can be exported to a file using the Save Text As command in the File menu. Text is saved as an RTF file. The entire text component (including embedded formulas or artwork) is exported, except for control information (such as link location and destination. (Hyperlink controls are interactive data that cannot be represented effectively outside of the CASHE context.)

Printing from the TextViewer sends the entire text contents (except hyperlink controls) to the printer. Context information, including the document citation, entry number, and entry title, are included whenever a text component is copied to the Clipboard, saved, or printed.

A string search can be performed on the window contents using the Find and Find Again commands in the Search menu. If the search term is

found, the window will be scrolled to the next occurrence of the term, and the term will be highlighted.

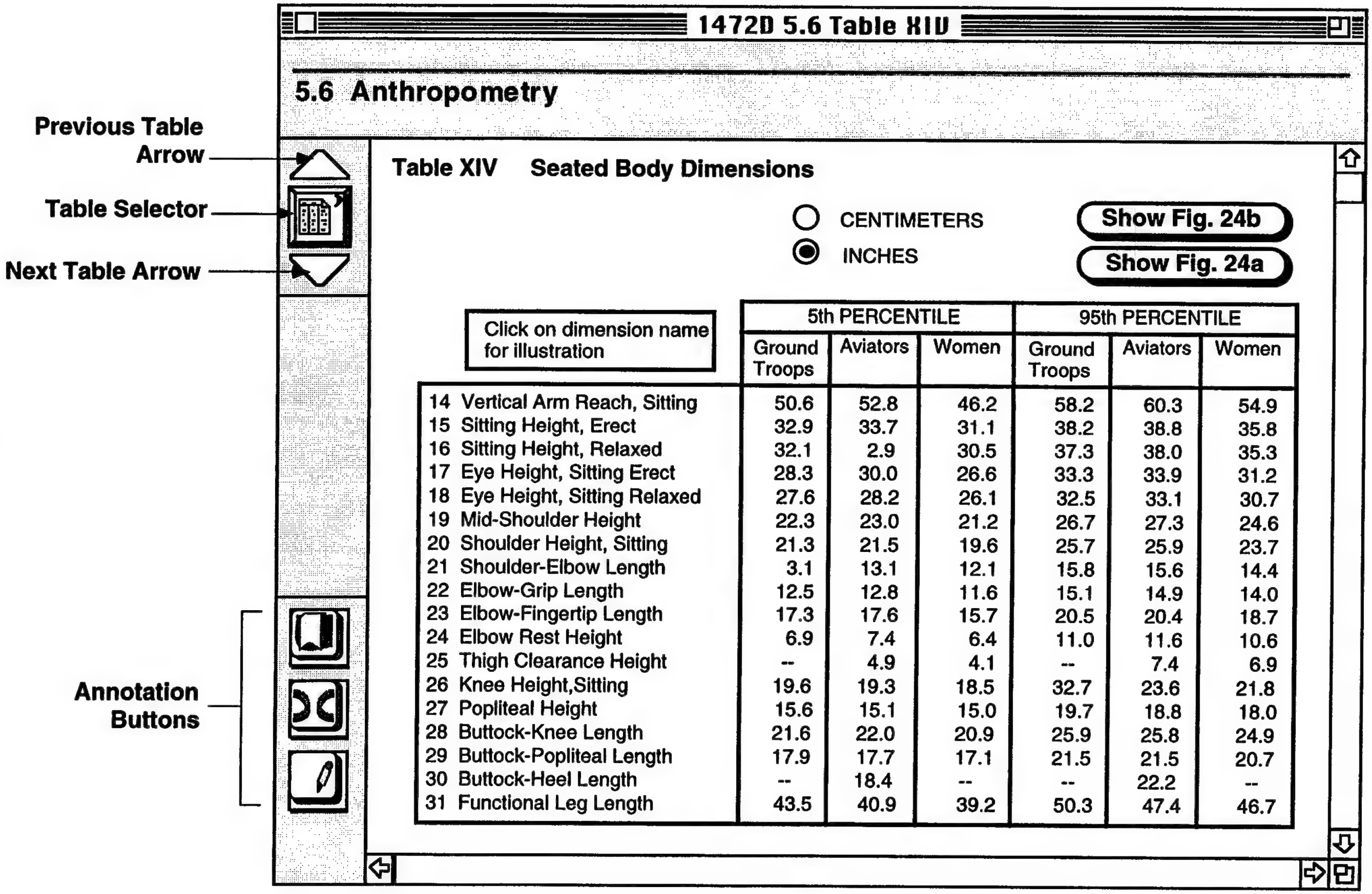
More than one TextViewer window may be open on the desktop at one time, as long as each window contains the text component of a different entry. When a text component is first opened in the TextViewer, the text is scrolled to the very beginning of the text component (its default display). When an open but inactive TextViewer window is reactivated by clicking it, selecting it from the Windows menu, or re-selecting the text component from the Entry Palette, then the TextViewer is displayed in its most recent state. In special cases (e.g., when the TextViewer is opened in response to a query search or is made active via a Glossary link or a reference link), the text will be scrolled to some pre-determined position.

Figure A-1 in Appendix A summarizes the control flow for the functions available in the TextViewer window.

3.2 TableViewer

The TableViewer is the component viewer for tables. Each table of an entry is a separate component. Most tables consist of a single panel; however, a few are composed of multiple panels joined by hypertext links.

Each table panel includes a title region and the table text proper. The title region contains the table title plus any embedded buttons that may be used to select different panels or representations of the table data. The table text region is subdivided into three areas: column headers, row stubs, and table body. The column headers provide identification labels for the columns. The row stubs (the leftmost column of the table) provide identification labels for the rows. The body of the table is composed of cells formed by the intersection of rows and columns. Headers, stubs, or body cells may contain text or graphics. Most CASHE tables have footnotes and/or credit lines. These are displayed at the bottom of the table, after the last table row.



	5th PERCENTILE			95th PERCENTILE		
	Ground Troops	Aviators	Women	Ground Troops	Aviators	Women
14 Vertical Arm Reach, Sitting	50.6	52.8	46.2	58.2	60.3	54.9
15 Sitting Height, Erect	32.9	33.7	31.1	38.2	38.8	35.8
16 Sitting Height, Relaxed	32.1	2.9	30.5	37.3	38.0	35.3
17 Eye Height, Sitting Erect	28.3	30.0	26.6	33.3	33.9	31.2
18 Eye Height, Sitting Relaxed	27.6	28.2	26.1	32.5	33.1	30.7
19 Mid-Shoulder Height	22.3	23.0	21.2	26.7	27.3	24.6
20 Shoulder Height, Sitting	21.3	21.5	19.6	25.7	25.9	23.7
21 Shoulder-Elbow Length	3.1	13.1	12.1	15.8	15.6	14.4
22 Elbow-Grip Length	12.5	12.8	11.6	15.1	14.9	14.0
23 Elbow-Fingertip Length	17.3	17.6	15.7	20.5	20.4	18.7
24 Elbow Rest Height	6.9	7.4	6.4	11.0	11.6	10.6
25 Thigh Clearance Height	--	4.9	4.1	--	7.4	6.9
26 Knee Height,Sitting	19.6	19.3	18.5	32.7	23.6	21.8
27 Popliteal Height	15.6	15.1	15.0	19.7	18.8	18.0
28 Buttock-Knee Length	21.6	22.0	20.9	25.9	25.8	24.9
29 Buttock-Popliteal Length	17.9	17.7	17.1	21.5	21.5	20.7
30 Buttock-Heel Length	--	18.4	--	--	22.2	--
31 Functional Leg Length	43.5	40.9	39.2	50.3	47.4	46.7

Figure 4. Sample TableViewer window

Figure 4 shows a sample TableViewer window. When utilizing a TableViewer, the user may:

- Select and jump to any table associated with the current entry;
- Scroll the table horizontally and vertically using the standard Macintosh scroll bar features;
- Resize the TableViewer window (which may cause the table to be clipped if the size of the table exceeds the new size of the content region);
- Navigate using hyperlinks embedded within any of the table regions;
- Access annotation features;
- Print the table;
- Copy the table or export it to a file;
- Search the text in the table headings and body for any text string;
- Open an associated animation or audio demonstration.

The sidebar controls on the TableViewer allow the user to navigate easily to other table components. Clicking the Table Selector button at the top left of the sidebar opens a pop-up menu listing all the table components of the current entry. Selecting a new table component opens the selected table in a new TableViewer window in front of the existing TableViewer window (which is left open). Clicking the Previous Table (upward-facing) arrow opens the table that immediately precedes the current table in the entry in numerical order (e.g., if the user is viewing Table 2, Table 1 is opened). Clicking the Next Table (downward-facing) arrow opens the table that immediately follows the current table in the entry. The previous or next table is opened in a new TableViewer window and the current TableViewer window is closed.

Using the annotation buttons on the lower left sidebar, users can view or create bookmarks, hypertext links, and notes attached to the table component currently displayed in the TableViewer.

The scrolling operation of the TableViewer is different from that of the other component viewers. Only the table text region scrolls. The table title region (including any embedded buttons), as well as the entry number and entry title, always remain in view. When the table is scrolled vertically, the row stubs move in synchrony with the table body while the column headers remain stationary. When the table is scrolled horizontally, the column headers move with the table body and the row stubs remain stationary. This method of "freeze-pane" scrolling preserves the relevant row or column labels when the body of the table moves, so the user is always able to interpret the table content.

TableViewer windows are moveable and resizable. When the TableViewer is resized, the viewport is repainted to reveal as much of the table content as possible. To help reduce user disorientation when resizing, the user's position within the table remains fixed and the table is clipped along the bottom and right sides. As a result, some rows or columns may be cut off if the new dimensions of the window content region do not coincide exactly with a row or column boundary.

The entire table can be exported to a file using the Save Table As command in the File menu. If the table has more than one panel, only the active panel is exported. The table is saved in tabbed text format.

The active table panel (except for controls) can also be printed. When the table is printed or saved to a file, footnotes and permission credits are always included, along with the entry number and title, table title, and document citation.

Copyright holders of some tables gave permission for the tables to be displayed on screen in CASHE but not to be exported or printed. When such a table is displayed in the active TableViewer window, saving and printing functions will be disabled and the Save Table As and Print commands in the File menu will be grayed out.

Tables with multiple panels have hyperlinks allowing users to navigate among panels. In some tables, some cells have hyperlinks to graphic

“pop-ups” that provide supplementary information about the cell contents. A prompt box identifies cells that have associated graphics. Tables may also contain embedded hyperlinks to other table, text, or figure components in the same or other entries.

Users can search the table text for a specified term or phrase by selecting the Find or Find Again command in the Search menu. If the search term is found, the window will scroll to and highlight the next occurrence of the term.

Some tables have associated audio or animation demonstrations that provide dynamic presentations to enhance the information in the table. The demonstrations are accessed through an embedded on-screen prompt that serves as a link marker. Clicking the prompt opens an animation window from which the demonstration can be launched.

Tables are independent components of an entry and can be opened and displayed whether or not the text for the given entry is open. Multiple TableViewer windows may be open at the same time if they contain different tables. When a table component is first opened in the TableViewer, the text is scrolled to the very beginning of the table (its default display). The exception is when the TableViewer is opened in response to a query search. In this case, the table will be scrolled to the first occurrence of the search term. When an open but inactive TableViewer window is again made the active window (e.g., by clicking it), then the TableViewer is displayed in its most recent state.

Figure A-2 in Appendix A summarizes the control flow for the functions available in the TableViewer window.

3.3 FigureViewer

The FigureViewer is the viewer for presenting complex graphical data, such as charts, diagrams, statistical graphs, and line drawings. Each figure in an entry is a separate component. A figure may have a single base panel or may be composed of multiple base panels accessible by embedded

buttons. Any base panel may have one or more overlays that can be displayed separately or together. (Refer to Figure 1 for a diagram showing figure component structure.)

An example of a FigureViewer window is shown in Figure 5. When using a FigureViewer, the user may:

- Select and jump to any figure associated with the current entry;
- Scroll the figure horizontally and vertically using the standard Macintosh scroll bar features;
- Resize the window, which causes the figure to be clipped if the size of the figure exceeds the new size of the content region;
- Zoom in or out on the figure to enlarge it or reduce it from the normal default size;
- Select one or more overlays for display from the set associated with the particular figure being viewed;
- Navigate to other figure panels or to other components using hyperlinks embedded in the figure;
- Display the figure caption;
- Access annotation features;
- Print the figure with the overlays currently shown;
- Copy the visible figure panel and overlays to the Clipboard or export them to a file;
- Search the caption for any text string; Open an associated animation or audio demonstration.
- Launch the DataDigitizer to digitize data points in the current figure panel.

The sidebar controls on the FigureViewer operate similarly to those on the TableViewer to allow the user to navigate easily to other figure components. Clicking the Figure Selector button at the top of the sidebar opens

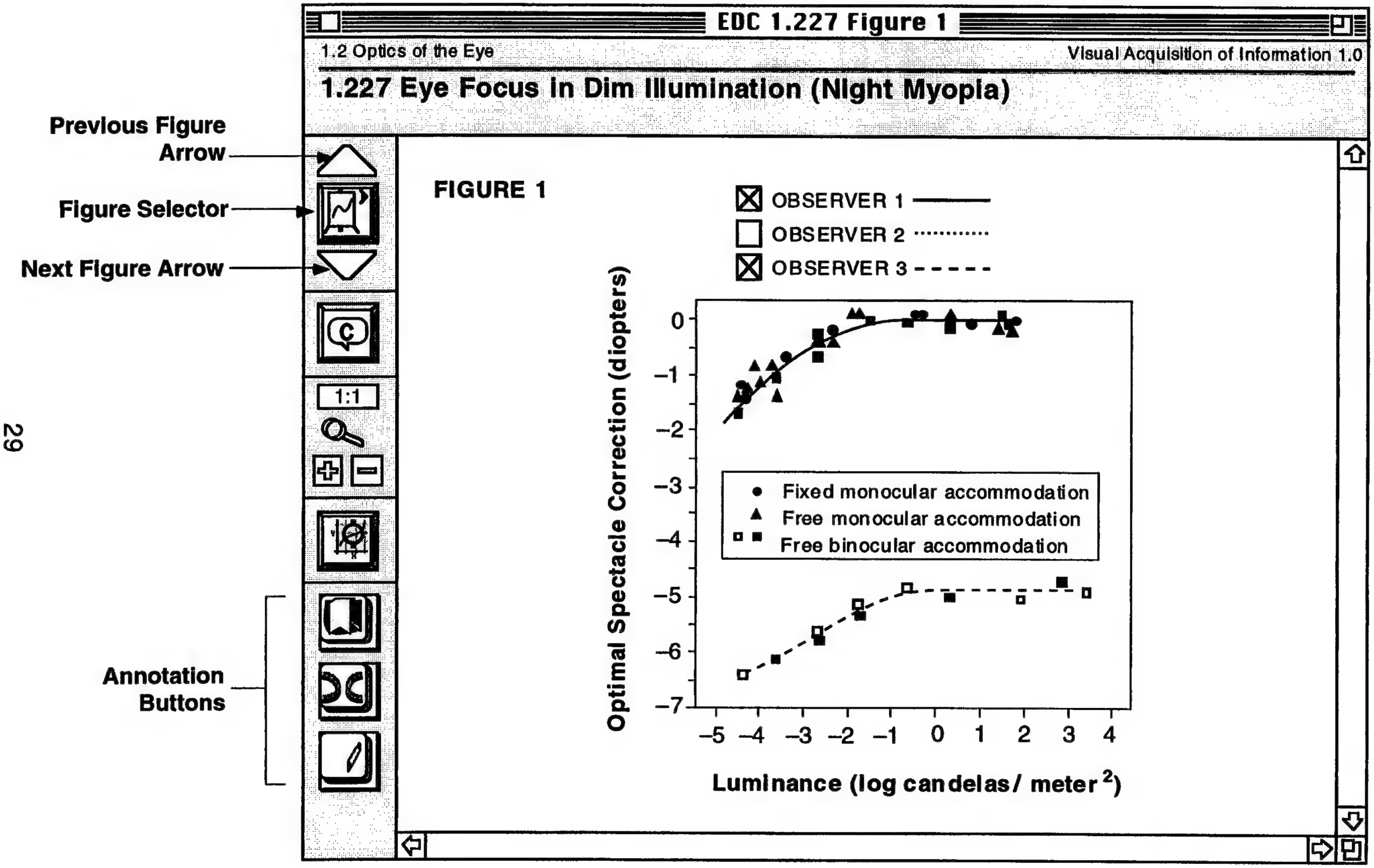


Figure 5. Sample FigureViewer window

a pop-up menu of all available figures for the current entry, from which the user can navigate to any figure. The new figure will be opened in a new FigureViewer window on top of the current window. Previous Figure and Next Figure arrow controls allow the user to page through all the figures in the entry in numerical order. Selecting a new figure component from these controls closes the current FigureViewer window and opens the selected figure in a new window.

Clicking the Caption button opens a caption window containing the caption and source for the current figure. Any credits or copyright notices that apply to the figure are included as part of the caption.

The FigureViewer provides zoom controls ("+" and "–") that allow users to enlarge or reduce the size of the figures. When the user selects a zoom operation, the mouse cursor changes to a magnifying glass. The user can click anywhere in the figure window's content region to select that point as the zoom focal point. The selected point moves to the center of the window content region as the figure is scaled. A readout indicates the current magnification ratio. Figures can be enlarged to 2x, 3x, or 4x, or reduced to 1/2x, 1/3x, or 1/4x. (The upper and lower bounds of the list were selected to just exceed the limits of figure usability.)

The FigureViewer window also provides access to the DataDigitizer, which allows the user to digitize and export figure data in order to explore analytic relationships in the graph. Clicking the DataDigitizer button on the viewer sidebar opens a DataDigitizer window containing the visible graphic contents of the active FigureViewer window. The features and operation of the DataDigitizer are described in section 6.3.

The annotation buttons on the lower left sidebar allow users to view or create bookmarks, hypertext links, and notes attached to the figure component currently displayed in the FigureViewer.

Window resizing and scrolling affect the figure in the same way as in many other Macintosh applications. When the user resizes the FigureViewer

window, the figure will be clipped on the bottom and/or right sides without regard to content if the size of the figure exceeds the current size of the window content region.

When a figure has more than one base panel, or when a base panel has overlays, there will be controls in the content region for displaying the various overlays or for moving from one base panel to another (such as the radio buttons shown in Fig. 5). More than one overlay may be present at once, but only a single base panel can be displayed in the viewer at any given time. Both overlays and base panels may contain links. Links embedded in overlays are not accessible when the overlay that contains them is not being displayed.

The figure in the active FigureViewer window can be copied to the Clipboard. The figure can also be exported to a file using the Save Figure As command in the File menu on the menu bar. Only the base panel and overlays that are currently visible will be copied or exported to a file. The entire contents of the figure is exported, including regions outside the current viewer boundaries. The graphic is saved at its original size, without regard to the current zoom factor. The figure is saved as a PICT file and includes only attributes that are representable in the PICT format.

The figure in the active FigureViewer can also be printed. The complete contents of the base panel and all overlays currently visible in the viewer are printed. The figure number, figure caption, permission credit line, document citation, and entry title are included whenever the figure is printed, copied, or saved to a file.

Copyright holders of some figures allow the figure to be displayed on screen in CASHE but did not give permission for the figure to be exported or printed. When such a figure is displayed in the active FigureViewer window, saving and printing functions will be disabled and the Save Figure As and Print commands in the File menu will be grayed out.

Users can utilize the Find and Find Again commands in the Search menu to perform a string search of the figure component for a specified term

or phrase. Only the figure caption will be searched. Invoking the Find command opens the caption window (or brings it to the front of the desktop if it is already open). If the search term is found, the caption will be scrolled to the first occurrence of the term and the term will be highlighted.

Some figures have associated audio or animation demonstrations. These demonstrations provide dynamic presentations to enhance the data in the current figure. The availability of a demonstration is signaled by a special icon in the content area. Clicking the icon opens an animation window from which the demonstration can be launched.

Figures are independent components of an entry and can be opened and displayed whether or not the text for that entry is open. More than one FigureViewer may be open on the desktop at a time, as long as each FigureViewer window contains a different figure component. Every multiple-panel figure or figure with overlays has a default configuration that specifies which panel and/or overlay(s) is to be displayed when the figure is first opened. When an open but inactive FigureViewer window is reactivated (such as by selecting it from the Windows menu), then the FigureViewer is displayed in its most recent state.

Figure A-3 in Appendix A summarizes the control flow for the functions available in the FigureViewer window.

3.4 Entry Palette

The Entry Palette (Fig. 6) gives continuity to an electronic entry. The palette provides access to all the text, table, and figure components that comprise the active entry, as well as to test benches related to the entry topic. The palette is always visible whenever an entry is open (i.e., whenever a component viewer is open). Although multiple viewers displaying text, tables, or figures from several different entries may be open simultaneously, there is only one Entry Palette and it always reflects the entry associated with the active

window. The document and entry number of the active entry are shown on the palette.

The Entry Palette contains navigational icons that provide an overview of the contents of the active entry and allow users to jump directly to any text, figure, or table component in the entry. Since designers tend to prefer graphic information portrayal to discursive text, the palette icons give equal prominence to text, figures, and tables. Clicking one of the icons displays a pop-up menu of all available components of that type in the entry. Users can scan these component menus to find the component most likely to contain the information they seek. (The table and/or figure palette icons are grayed out if the entry has no figures or tables.)

Selecting the entry title from the pop-up menu under the Text button opens the text component of the entry in a TextViewer window. Selecting a figure or table component from the pop-up menu under the Figure or Table

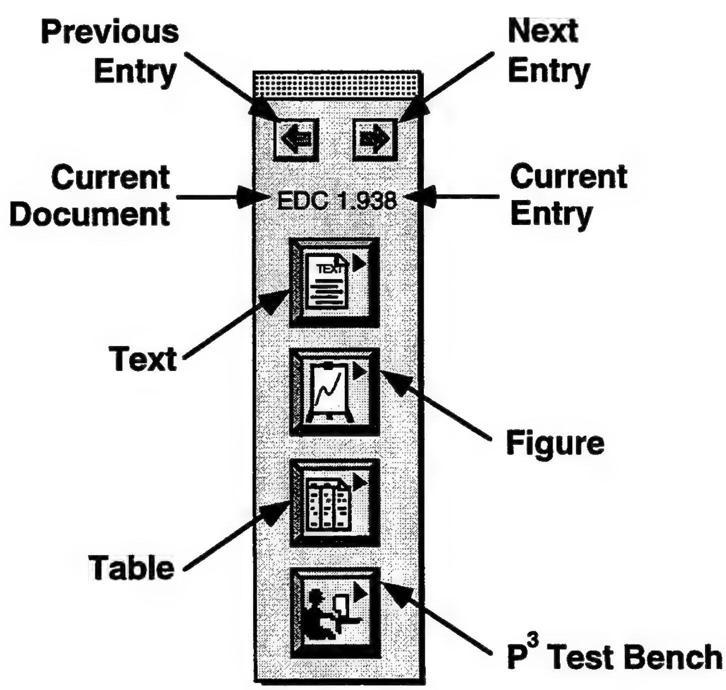


Figure 6. Entry Palette

button opens the corresponding figure or table in a new FigureViewer or TableViewer window with the default configuration for that component. If additional figure or table selections are made from the menu lists under the Figure and Table buttons, the new figures or tables will be opened in new viewer windows on top of the existing FigureViewer or TableViewer windows.

Selecting a test bench from the pop-up menu under the P^3 Test Bench button launches the corresponding test bench application. If the test bench has multiple topics, the topic that best illustrates the subject of the current entry is flagged in the opening window of the test bench. (The P^3 Test Bench button is grayed out if there are no test benches associated with the entry.)

The Entry Palette also contains two arrow buttons that allow users to browse adjacent entries. Clicking the Previous Entry button (left-facing arrow) opens the text component of the previous entry (in numerical order). Clicking the Next Entry button (right-facing arrow) opens the text component of the following entry. When a new entry is opened using the arrow buttons, any open components of the current entry are automatically closed.

Figure A-4 in Appendix A summarizes the control flow for the functions available on the Entry Palette.

4. INFORMATION ACCESS IN THE REFERENCE DATABASE

One of the important advantages of computerizing the CASHE database is that it makes it possible to use rapid and powerful search and retrieval techniques and to support several different information access modes in parallel. In CASHE, there are three primary means of accessing information: browsable outlines, text search, and hyperlinks.

4.1 Browsable Outlines

Users of printed books often learn about their structure and contents and find the information they need by examining the Table of Contents. The Table of Contents is, in effect, a browsing outline, one that arranges the topics covered in the book in a hierarchical order. Such an outline can be computerized to serve a similar access function. Just as the headings in a printed Table of Contents carry page numbers to allow the reader to find them in the printed book, the headings in a computer-based Table of Contents can be linked with the corresponding locations in the electronic text.

There are two ways in which a computer-based outline such as a Table of Contents can be more efficient to use than a printed one, however. First, a computerized outline can be programmed to show any desired level of detail, from only the most abstract levels, such as a list of section titles, to a comprehensive hierarchy that also shows the chapter titles, subdivisions within chapters, and headings nested beneath them. Since the user can expand just those headings that are of interest, he or she can rapidly pinpoint the appropriate place at which to enter the text. Second, once the desired starting point is determined, the computerized link between a heading and the corresponding text enables that text to be retrieved and displayed quickly.

In addition to the Table of Contents, many of the other reference aids often found in books, such as a back-of-the-book index and lists of figures or tables, can be treated on-line as outline hierarchies to help users rapidly find information.

CASHE provides several such electronic outlines—some converted from the printed documents and some created specifically for the database—to help users explore the reference documents and locate the information they need.

Each browsing outline provides a structured listing of topics covered in the reference database, but approaches the subject matter from a slightly different organizational perspective. In a sense, the outlines serve as complementary filters on the database information. Users can select the filter that most closely matches the way they have structured their queries.

This flexibility is important, since the designers who use CASHE will have varying backgrounds and interests, and will come to the database with different problems and information goals. Because each of the access outlines approaches the database information from a slightly different perspective, users are more likely to find an approach that matches the way they have framed their questions and thus are more likely to find information that meets their needs.

4.1.1 Integrated Outline

The Integrated Outline is a multilevel taxonomy that offers a comprehensive view into the combined contents of both the *EDC* and *MIL-STD-1472D*. The Integrated Outline is a powerful access tool that makes it easier for users to find information in a multidocument database. Because the outline covers the entire database, users do not have to guess which document is most likely to contain the information they need. They can simply locate the topic of interest, and the outline will direct them to entries in the appropriate document. As new documents are added to the CASHE database, these can be

easily mapped onto the Integrated Outline, so that users will always have a single point of entry into the entire reference database.

A second characteristic that makes the Integrated Outline a powerful access tool is that it combines both subject-oriented and applied perspectives on the database information. The top level of the outline consists of six broad areas:

1. Human physical characteristics and capabilities
2. Human perceptual and performance capabilities
3. Display and control design
4. Equipment and vehicle design
5. Workspace design
6. Environment

The entries in the reference database are mapped repetitively onto both the first two sections, which have a behavioral orientation, and the last four sections, which take a more applied viewpoint on the information. This redundancy helps to accommodate individuals with different backgrounds and interests by allowing users to browse from the perspective that most closely matches their current needs and be assured of locating all pertinent information.

4.1.2 Tables of Contents

Users can browse a detailed Table of Contents for each document (converted from the printed version) to see how the document is structured in its traditional book form and access entries that strike their interest. The Table of Contents for the *EDC* has a uniform structure of three levels. The top level consists of the 12 broad subject areas covered in the *EDC*. These subject areas are decomposed into topic sections. Each topic section contains a number of entries addressing specific, narrow facets of the topic.

MIL-STD 1472D is organized as a series of numbered paragraphs in a nonuniform hierarchical arrangement to a maximum depth of six levels.

The *MIL-STD 1472D* Table of Contents includes the top four levels. The *MIL-STD 1472D* structural hierarchy creates problems for browser construction, because text is not always at the lowest level of the hierarchy. Some third-level headings have an associated text paragraph but also have fourth-level subheadings below them. This creates ambiguity for a link from the heading—should it display the text associated with that heading or open the lower-level subheads? This problem is circumvented in the CASHE outline browser by expanding the third and fourth levels of the hierarchy simultaneously.

Several additional contents-related browsing outlines are available for *MIL-STD-1472D*. A Table of Entries outline allows users to access the document according to the entries into which *MIL-STD-1472D* has been subdivided. These entries are logical groupings of the information in *MIL-STD-1472D* created to make the document easier to use in the CASHE environment. Figure and table lists are also available for *MIL-STD 1472D* to provide direct access to graphic or tabular components (such an approach is not feasible for the *EDC* because of the large number of figures and tables).

4.1.3 Indexes

Users with a particular concept or subject in mind can browse the alphabetical index for each document, which has been reproduced from the printed edition. Although the back-of-the-book indexes somewhat duplicate the functions of a full-text machine search (which CASHE also supports), they are retained in the electronic version because they reflect a human interpretation and organization of the subject matter and provide subheadings and cross references that help users focus their search more efficiently.

The first level of each index is the letters of the alphabet. Expanding a letter shows the list of index terms for that letter. Many of these terms may be expanded further to lower-level subheadings.

The *EDC* Index has approximately 2000 top-level index headings. Most incorporate several subheadings, and many of these contain another

nested level or more, making about 10,000 headings in all. Headings are indexed to relevant *EDC* entries by entry number and are hyperlinked to the text component of the entry.

MIL-STD-1472D has approximately 5000 index headings. Index headings reference the paragraph in which the term is found. The index appears to have been created largely by permutation of terms in paragraph headings.

4.1.4 Outline Viewer

All access outlines are presented in an OutlineViewer that facilitates browsing and supports the heading manipulation and hypertext linking that make electronic outlines so powerful. The headings at the lowest level of an outline (which generally consist of entry numbers and/or entry titles) link directly to the text components of entries in the *EDC* or *MIL-STD-1472D* reference documents. For this reason, the lowest outline level is called the "anchor level." The presentation style of each heading is controlled by an associated internal style sheet for outlines, which defines such aspects as the amount of indentation for each sublevel and distinctive rendering to differentiate link-level headings from higher-level headings.

Figure 7 shows a sample OutlineViewer window. When employing an OutlineViewer, the user may:

- Selectively show and hide parts of the outline hierarchy;
- Navigate to other components by activating links attached to the lowest level line items;
- Vertically scroll the outline using standard Macintosh scroll bar features;
- Resize the window, which causes the outline to be reflowed into the content region;
- Perform a string search on the contents of the outline window.

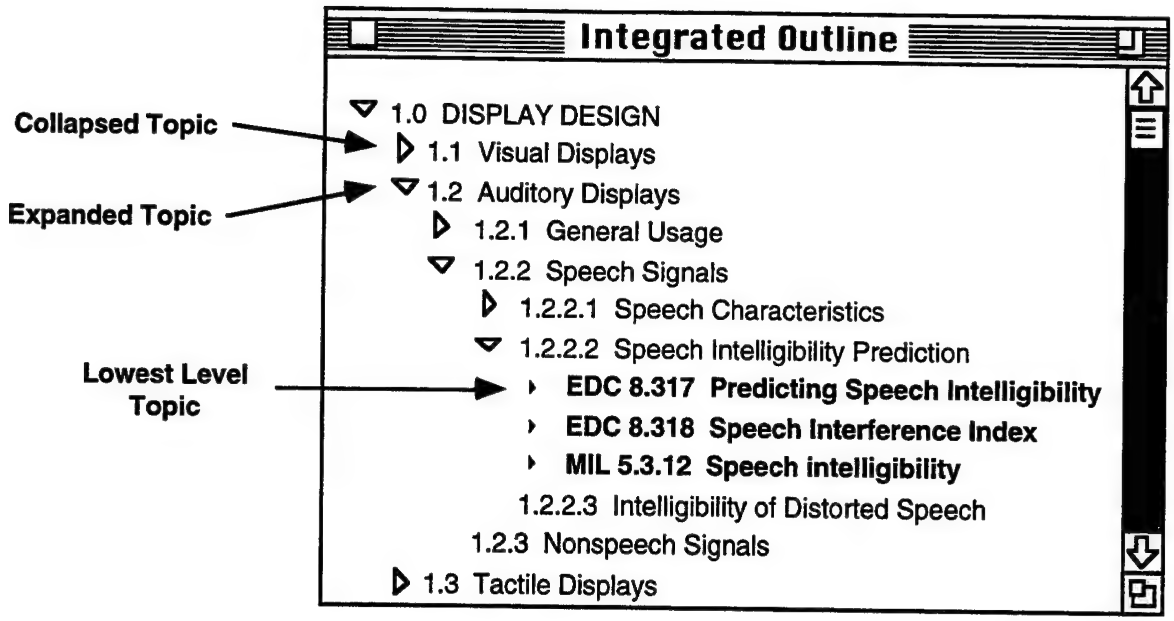


Figure 7. Sample OutlineViewer window. The window contains a portion of the Integrated Outline.

Outlines are initially displayed in a pre-determined default configuration (generally with only the highest level headings displayed). Headings are expanded or collapsed by clicking the hollow triangles that appear to the left of each line (similar to the way directories are expanded and collapsed in the Macintosh Finder). Right-facing hollow triangles indicate topics that may be expanded, while downward-facing hollow triangles signal topics that are already fully displayed. Filled triangles indicate topics at the lowest (anchor) level. The anchor-level headings are rendered as bold text, to indicate that they are hyperlinks. Clicking an anchor-level heading navigates the user to the destination component. Unless the destination component is already open, it will appear in its default configuration. If the destination component is already open, then it will simply be made active in its current state.

To make locating topics in the outlines even easier, users can perform simple text-string searches on the contents of the OutlineViewer using the Find and Find Again commands in the Search menu on the menu

bar. The Find/Find Again commands will search the entire outline within the viewer window, including unexpanded headings.

OutlineViewer windows are moveable and resizable. If the window is resized, the text is reflowed to fill the window. Outlines may be scrolled vertically to reveal additional text above or below the viewing region. OutlineViewers retain their state (including last selected item) as long as they appear on the desktop.

4.2 Text Search

Users who have defined their object of search in terms of a specific word or phrase can rapidly access the information they need by performing an electronic full-text search on the reference documents in the database. Users can conduct global searches of the entire reference database for single terms or for query expressions containing Boolean operators using the Query function. They can also perform local string searches of the content of the active viewer window using the Find function. Both search functions can be accessed from the Search menu on the main menu bar.

4.2.1 Global Search Using the Query Function

The Query command and related functions in the Search menu allow users to perform a global search of all database documents for specific terms or phrases in entry text. The Query function searches all text, figure, and table components in both the *EDC* and *MIL-STD-1472D*. For figure components, Query searches the figure captions for the query term. Browsing outlines, user annotations, test benches, and context information (such as the entry title of each component and *EDC* section and subsection titles) are not searched by the Query function.

Many other computer applications perform an incremental or an iterative search. Both the incremental and the iterative search processes are single-phase processes: after the user issues a query, the system navigates

directly to the next text location that satisfies the query. The user must visit each and every matching text location in order of occurrence until a relevant location is found. Finding the desired information can be a daunting task when the document or database is large and there are a large number of matches to the search term.

CASHE provides a two-phase search process to help the user find relevant material more efficiently. Rather than navigating the user directly to the first match, the system compiles a list of all "hits" (i.e., all entry components containing text that satisfies the search query). The user inspects this list and sequentially selects one or more "hits" to examine.

Such a two-phase search process is particularly advantageous when the database to be searched is stored on a CD-ROM, as with the CASHE documents. The CD-ROM is a very slow device, and the user may be required to wait for a noticeable amount of time during each access. By reducing the number of CD-ROM accesses required to locate relevant text, the two-phase process both saves time and avoids making the user feel penalized for performing searches.

4.2.1.1 Query Structure: When the Query command is selected from the Search menu, a Query dialog box opens (see Fig. 8). Users enter the desired search expression into the text field of the Query dialog box. Queries are case insensitive. Words of any length may be entered, but only the first eight characters of a word are used to find matches. Searches are limited to alphabetic characters only (plus parentheses for grouping).

Searching only for isolated terms or literal text phrases becomes very limiting in large databases like CASHE. Such searches either produce too many "false alarms" or miss important instances where the text does not take the exact form of the query. To provide more flexibility, CASHE utilizes a query language that is similar to those employed in many search services, but that is easier to use.

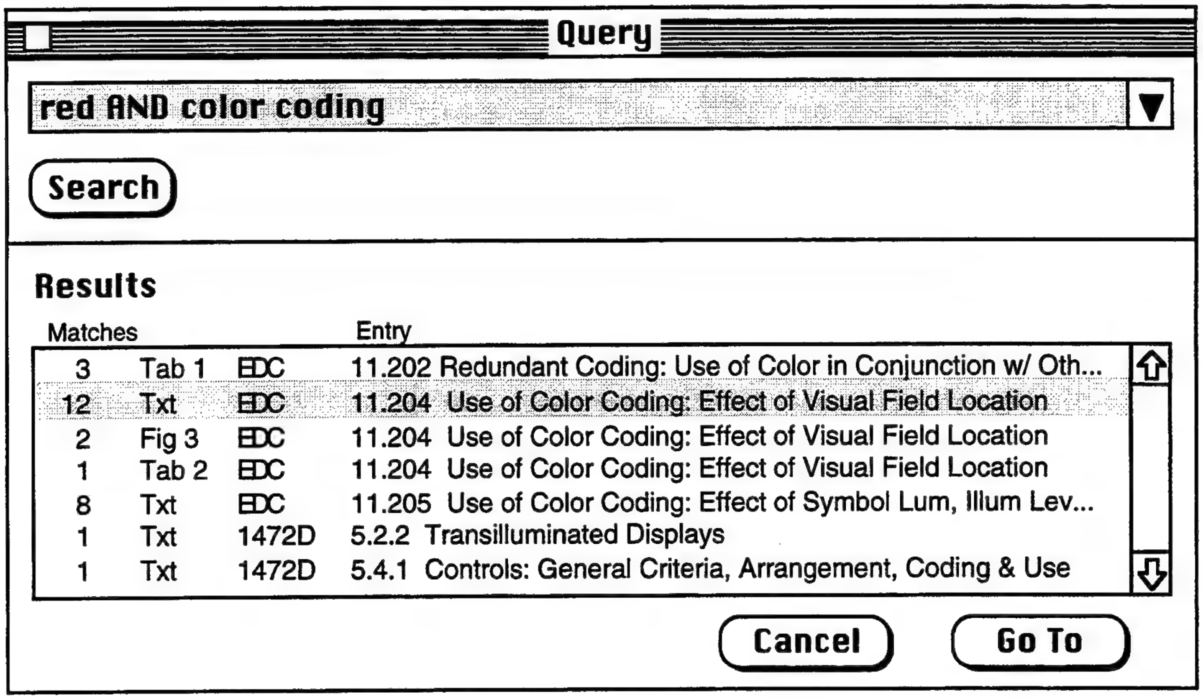


Figure 8. Query dialog box

The CASHE query language is simple yet powerful. It uses only three logical operators: AND, OR, and ADJ. The AND and OR operators are intuitive. For example, a query of the form "stimulus AND control" finds all entries that contain *both* the word *stimulus* and the word *control*. Queries of the form "stimulus OR control" find all entries containing the word *stimulus,* plus all entries containing the word *control*. ADJ indicates adjacency and also defines the order of terms. For example, the search query "color ADJ coding" will find all entries containing the phrase *color coding* (but not entries that contain the phrase *coding color* or entries in which the words *color* and *coding* are separated by intervening words). A space between words implies the ADJ operator (i.e., "color coding" is equivalent to "color ADJ coding"), since most people find an expression like "color coding" more natural.

Parentheses may be used for grouping and operate just as in mathematics to control the order in which operations are executed. For example, because AND has a higher precedence than OR, a query such as "visual OR auditory AND adaptation" will produce a list of all components that

contain both the term *auditory* and the term *adaptation*, plus all components that contain the term *visual*. If, on the other hand, the query is written as "(visual OR auditory) AND adaptation," the result is a list of all the components that contain the term *adaptation* and also contain either the term *visual* or the term *auditory* (or both).

AND, OR, and ADJ are rendered in upper case when they are entered into a query expression in the Query box. Since these three words are operators, users may not search for them in the database. Frequently used words such as *the, of, by*, etc., are omitted from the search indexes and may not be located using the Query command. Entering such "stop" words as query terms will return a result of "No entries matched."

A formal specification of the query language, a full list of "stop" words, and other technical details regarding full-text search are given in sections 8.5.2 and 9.3.

4.2.1.2 Results Retrieval: Once the user has specified a query, the system conducts a global search through all text, table, and figure components of all database documents for strings that satisfy the query.

A status bar in the Query dialog box indicates when the search has been completed and reports the total number of hits found for the query. Components that contain one or more matches to the query are displayed in the scrolling results field of the Query window. If no match is found, the status bar will read "No entries matched" and the results field will be empty.

Components that contain query matches are presented in the results field in order by entry number, with the entries from the *EDC* given first, followed by the *MIL-STD-1472D* matches. The results field display lists: (1) the number of times a match to the query string was found in that component; (2) the type of component (text, table, or figure); (3) the document (*EDC* or *MIL-STD-1472D*); (4) the entry number; and (5) the short entry title. Presenting this information about each query match makes search more efficient for the user

by providing a means of judging whether or not a given component is likely to contain information of interest.

Users can navigate directly to a desired match on the results list by double-clicking the corresponding results line or by selecting the line and clicking the Go To button. The selected component will be opened, and the text will be scrolled to the first occurrence of the query term with the term highlighted. Selecting the Next Match command from the Search menu scrolls the text in the current viewer window to the next occurrence of the query string. If the text is already scrolled to the last occurrence of the search term in that window, the next entry component in the results list will be opened and scrolled to the first occurrence of the query string in that component. Selecting Previous Match from the Search menu scrolls the text in the current window to the previous occurrence of the query string. If the user is at the first occurrence in that window, then the previous entry component in the results list will be opened, scrolled to the last match in that component.

The results list from the most recent query will continue to be displayed in the Query dialog box until the user executes a new query or closes the dialog box. The user may return to the results list as often as desired to select a new component match for viewing by clicking in the Query dialog box or selecting it from the Windows menu to activate it. When the Query dialog box is reactivated, the last component match selected the last time the user accessed the list will be highlighted in the results field.

The user can step easily through the component matches in the results list using the Previous Component Match and Next Component Match commands in the Search menu. Previous Component Match opens the previous component listed in the results field and scrolls the viewer window to the first occurrence of the query term. Next Component Match opens the next component on the results list and scrolls the viewer window to the first occurrence of the query term.

When the user conducts a new Query search, the previous results list is overwritten and is no longer accessible, even if the new search does not result in any matching entries. However, the user may select any of the previous eight query expressions from a pull-down menu under the arrow to the right of the text entry field to reinstate that query expression in the query field. The query term may be edited as desired before being reissued.

Figure A-5 in Appendix A summarizes the control flow for the functions available in the Query dialog box.

4.2.2 Local Search Using the Find Function

The Find function operates locally and allows users to perform a simple string search for a word or phrase in the text of the active window (including the portion of the text scrolled out of view) rather than searching the entire contents of the database as with the Query command.

The Find command can be applied to the text in any TextViewer or TableViewer window, as well as to supplementary database material such as access outlines, Glossary, and Design Checklist. Find is especially useful to locate topics of interest in browsing outlines, since outlines are not searched by the Query function. If the Find command is invoked when a FigureViewer window is the active window, the associated caption will be opened and searched (the graphic itself will not be searched). Context information, such as the entry title and number, are not included in the search.

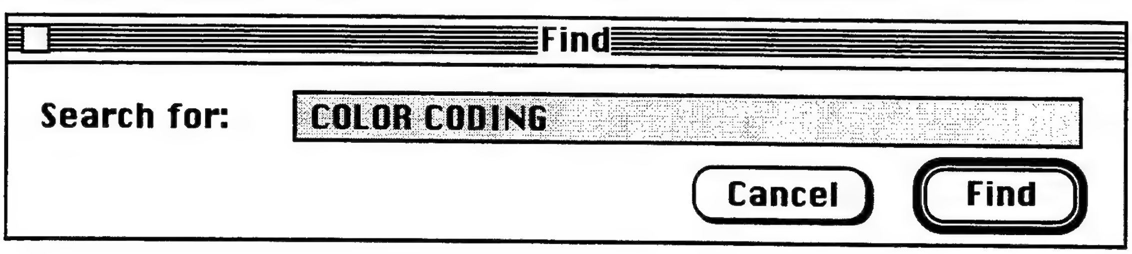


Figure 9. Find dialog box

The user enters the desired text string in the Find dialog box (Fig. 9). Find will scroll the window to the first occurrence of the search term and highlight the term. Selecting Find Again from the Search menu will scroll the window to the next occurrence of the term. When the end of the window text is reached, search will continue at the beginning of the text. If no match is found, a beep will sound. The search string may be any text the user can type at the keyboard. Word- or phrase-matching is case insensitive. Logical operators cannot be used in the search expression for the Find function.

4.3 Hyperlinks

One of the hallmarks of hypertext is that it supports electronic connections, or hyperlinks, between related items of information that allow users to jump quickly from one place to another in the database. Hyperlinks facilitate access by supporting a nonlinear mode of information search in which users can easily follow a particular train of thought or topic path.

There are two broad classes of hyperlinks in CASHE: preset and user-defined. Preset hyperlinks are "hard" (i.e., fixed) links embedded in the electronic documents; they reflect structural properties of the database and the views of the document's creator or a specialist in the subject matter regarding which elements are related to one another. User-defined hyperlinks are "soft" links created by the user to join two database components whose relationship is of personal relevance to the user. User-created hyperlinks are discussed in section 5.1.3. The remainder of this section deals exclusively with preset hyperlinks.

Every hyperlink has a source, or point of origination, and a destination, or the point to which the user will be transported when the hyperlink is activated. CASHE hyperlinks are unidirectional; that is, the user can navigate directly from the source to the destination but not from the destination back to the source. Every hyperlink source also has an anchor—a specific region ("hot spot") in the source component from which the hyperlink

is activated. Clicking the anchor immediately takes the user to the link destination.

Preset hyperlinks may occur in any component—text, table, or figure. In figure components, hyperlinks allow users to move easily between related figure panels, arrange overlays, or jump to related components. The hyperlinks are easily identified as buttons within the viewport. The button label or icon indicates the destination of the link. Some table components also have hyperlink buttons. These specialized figure and table hyperlinks are described in more detail in sections 8.2.1 and 8.3.1.

Several other types of hyperlink are built into the CASHE reference database. Both documents in the database have cross references embedded in the running text that point users to related entries or sections. Each *EDC* entry (and some *MIL-STD-1472D* entries) also have a separate Cross References section at the end of the entry listing other entries on the same topic. All cross references—both those embedded in the text and those in the separate Cross References section—are "hot" links that navigate the user directly to the referenced entry. When the user clicks the cross reference, the text component of the appropriate entry is opened. For *MIL-STD-1472D* cross references, which may refer to a numbered section that falls in the middle of an entry, the text is scrolled to bring the referenced section to the top of the viewer window.

One special enhancement of the CASHE database is interdocument linking—cross references joining related entries in *MIL-STD 1472D* and the *EDC*. For example, specific standards in *MIL-STD-1472D* may be linked to research data in the *EDC* that support or provide a rationale for them (or perhaps refute them!), while *EDC* data may be linked to practical guidelines in *MIL-STD-1472D* that provide a real-world perspective. Interdocument cross references are always placed in the Cross References section at the end of the entry.

These interdocument cross references were added by a subject-matter expert during document processing for the CD-ROM. Although

automated interdocument link generation was considered, it was ultimately decided that professional review of the documents for cross-referencing would provide higher-quality links of greater potential value to designers than cross-referencing based on word- or phrase-matching algorithms.

Entry text in the *EDC* often contains references to bibliographic sources listed in the Key References section of the entry. Each such reference is a hyperlink that provides direct access to full bibliographic information on the source by scrolling the text to the appropriate source citation. Figure and table references in the running text of both documents are also hyperlinks. Clicking an embedded figure or table number immediately opens the indicated graphic or tabular component for viewing.

To help users interpret specialized material, technical terms in the text are linked to their Glossary definitions. Defined terms are bolded in the text. Clicking a bolded term opens a Glossary window, scrolled to the term's definition.

Although Glossary terms are bolded in the text, other hyperlinks occurring in text components (and the body text of table components) do not have any special on-screen rendering. They can be recognized by their content and the context in which they occur. For example, all references to specific

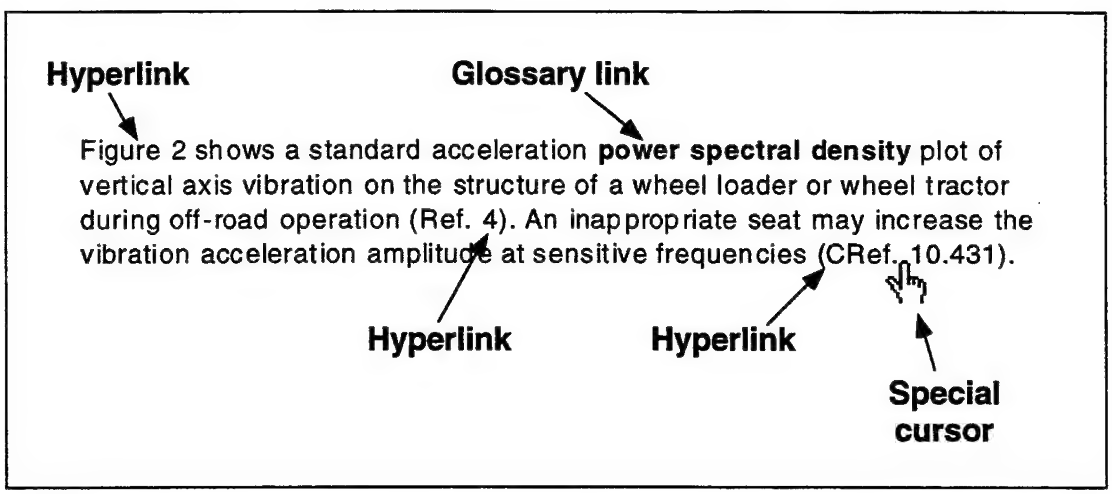


Figure 10. Examples of text hyperlinks. The normal arrow cursor changes to a pointing-hand cursor when it passes over a hyperlink.

figures, tables, Key References sources, or other entries (such as "Fig. 1," "Ref. 2," "CRef. 6.502," and "sect. 5.2") are hyperlinks, as are all items in the Cross References section. In addition, when the cursor is over a hyperlink, it will change from its normal arrow shape to a pointing hand. Figure 10 shows examples of hyperlinks and the pointing hand cursor.

4.4 Windows Menu

The Windows menu on the menu bar gives the user a quick overview of the information already present on the desktop and provides an easy means of accessing this information. Its functioning is consistent with the standard Macintosh human interface guidelines. The Windows menu provides the user with a list of all open CASHE windows. All types of windows (entry components, outlines, Glossary, audio/video demonstrations, etc.) are entered in the windows list. (Test bench windows do not appear in the Windows menu because test benches are separate applications—instead, they are listed in the Applications menu maintained by the operating system.)

Selecting a window in the Windows menu brings that window to the front of the desktop and makes it the active window. Windows appear in the menu list in order of their position on the desktop from front to back. Menu listings match the drag bar titles of the windows. Thus, for database entries, menu lines identify the document, entry number, and component (e.g., *EDC* 6.305 Figure 2).

4.5 History Menu

The History menu on the menu bar provides another means of accessing the database documents that follows the chronology of the user's current CASHE session. The History menu lists the 12 most recently visited entries in reverse chronological order. In effect, it provides a "bread crumb trail" (or automatic bookmarking facility) that lets users return to recently viewed data—for example, if they become lost after following hyperlinks or

decide to pursue another path of reading. Entries appear in the menu in reverse chronological order because the user is most likely to want to navigate to items that were more recently visited and that the user remembers. Items at the top of the list are the easiest to access.

Listings in the History menu identify the document as well as the entry number of the item. Selecting any entry in the History menu brings the window containing that entry to the front of the desktop (or opens the entry if the component is no longer on the desktop). Entries are entered in the History menu as text components, regardless of which components of the entry were open, and selecting an entry from the History menu always opens the text component of the entry.

Since the goal is to aid the user in navigating to specific database information, only document entries appear in the History menu. Test benches, access outlines, glossaries, etc., do not appear in the History list (although they *do* appear in the Windows menu).

The contents of the History menu are saved when the user saves the current session, so the user can restore the current context at a later time with a record of the entries visited (see sect. 5.2).

4.6 Bookmarks List

Users can create their own access route through database entries by placing bookmarks on components they have found to be especially relevant to the needs at hand. Users can then quickly return to any bookmarked component by selecting it from the Bookmarks list under the Annotation menu. This annotation function is discussed in more detail in section 5.1.1.

5. USER CUSTOMIZATION

CASHE provides several ways in which users can customize and document their interactions with the database. These include annotating the database with their own comments, place markers, and access routes; saving session work contexts; and exporting and importing selected material.

5.1 Annotations

Readers of printed manuals frequently write comments in the margin, insert bookmarks—and, in these modern times, attach sticky notes—to annotate the material for their personal use and make it easier to return to information they have found to be of value. CASHE supports electronic analogues of these operations that enable users to personalize the information in the reference database and customize their interactions with it.

Users can attach their own electronic notes to text, figures, or tables in any database entry. They can place electronic bookmarks that allow them to return rapidly to specified locations in the database. They can create hyperlinks between any two database components that allow them to navigate directly from one component to the other.

These annotation functions allow users to impose their own structure and access routes on the database information. Database usage can be tailored to current work needs, customized for several different projects at once, and changed when work focus shifts. At the same time, the annotations capability is structured in a way that avoids obscuring the material to be annotated and clearly distinguishes the user's additions from the original material.

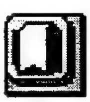
The three annotation functions—bookmarks, links, and notes—may be accessed from any of the component viewers (TextViewer, FigureViewer,

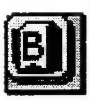
and TableViewer) using the annotation buttons provided. The annotation functions in the viewers are local and are associated with the entry component in the viewer window; they allow users to view, create, or delete annotations for that component. Figure A-6 in Appendix A summarizes the control flow for the functions in the annotation region of the viewers.

Annotation functions can also be accessed using the Annotations menu on the menu bar. The annotation functions in the Annotations menu are global; they allow users to view a list of all components that contain annotations created during a session or reloaded from a previous session.

5.1.1 Bookmarks

Bookmarks are electronic markers that allow users to return rapidly to a given component. A bookmark may be attached to any text, figure, or table component. A bookmark serves as an alternate name for a location in the information space. It enables users to retrace their steps easily to information they have found to be of value.





A bookmark is placed on or removed from the current component by clicking the Bookmark button in the annotation region at the bottom of the viewer window sidebar. Clicking the Bookmark button opens a moveable dialog box from which users can create and name a new bookmark or rename or remove an

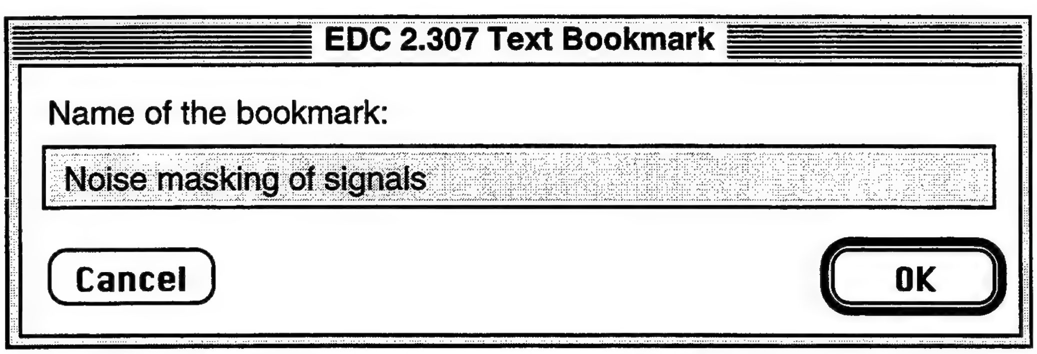


Figure 11. Dialog box for creating a bookmark

existing bookmark. Figure 11 shows the dialog box for creating a bookmark. If a bookmark has been placed in the current component, a "B" is displayed on the tag in the Bookmark button on the viewer sidebar.

Bookmarks operate at the component level. A bookmark may be placed on any text, table, or figure component; however, only one bookmark may exist for any given component. Bookmarks are preserved by saving the current session (saving sessions is explained in section 5.2).

The user can access a list of all bookmarks for the current session using the Annotations menu on the menu bar. Selecting Bookmarks from the Annotations menu opens a window showing all the bookmarks that have been placed during the session (see Fig. 12). The bookmarks are listed in the order in which they were created and show the number of the marked entry, the type of component (text, table, or figure), and the bookmark name. Any marked component can be opened by double-clicking the desired bookmark or selecting the bookmark and clicking Go.

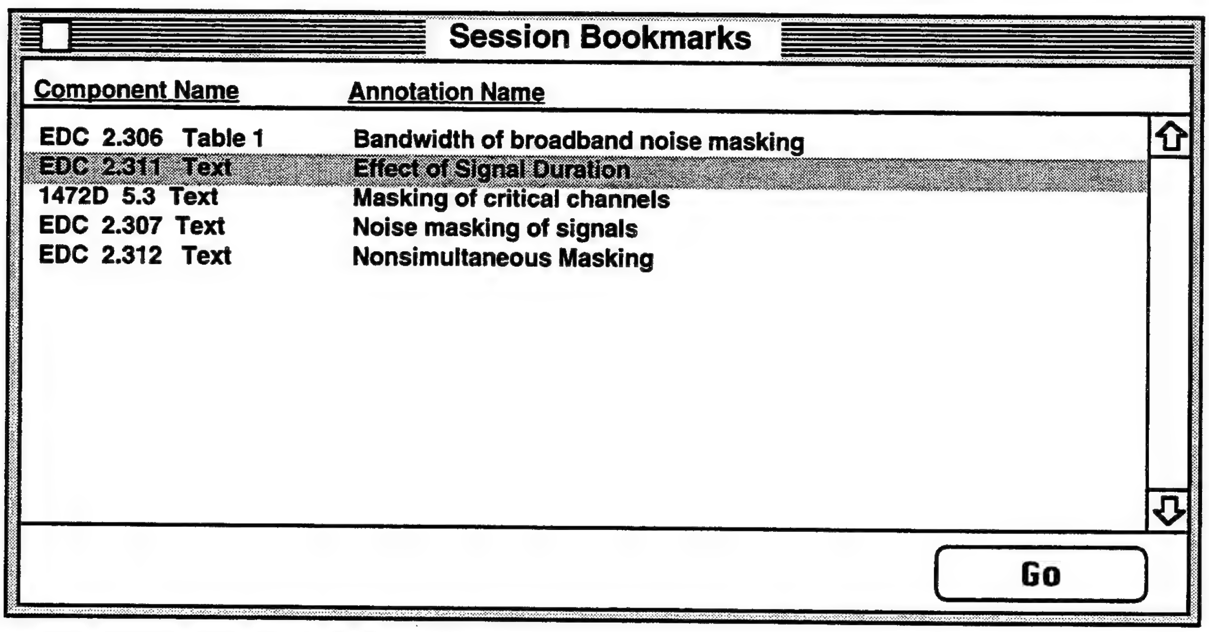
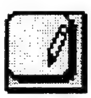


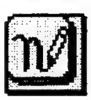
Figure 12. Session Bookmarks window showing all bookmarks created during the current session

The bookmark annotation feature allows the user to find information by using his or her own names for various components. Bookmarks can be especially effective for sporadic users who are human factors novices. Such users are likely to have different names for things, and they will not become expert in the operation of the CASHE tool until they have used it for several projects.

5.1.2 Notes

The Notes feature allows users to append additional information or commentary to a particular entry component, while at the same time keeping user material clearly differentiated from the original source material. A text note may be attached to any text, figure, or table component using a simple facility very similar to the Macintosh notepad.





The user Notes feature is accessed from the current component by clicking the Notes button in the Annotation region of the viewer window sidebar. Clicking Notes opens a moveable dialog box from which users can create a note, or view, edit, or remove an existing note. Each note can be given a name. The user creates notes via normal Macintosh editing. Notes appear in a single font, based on an internal style sheet that cannot be altered by the user. Attributes such as boldfacing or italics are not supported. The user may copy, cut, and paste text to or from the note by means of the Clipboard. Figure 13 shows the dialog box for creating a note. When there is a user note attached to the active entry component, an 'n' will appear next to the pencil on the Notes button on the sidebar of the viewer window.

Selecting Notes from the Annotations menu in the menu bar opens a window (Fig. 14) listing by name all the notes created during the current session, in order of the entry component to which the note is attached. Users can navigate to any note listed by double-clicking the desired note or selecting

the note, then clicking Go. If the note being opened is not attached to the active component, the appropriate component is also opened.

The dialog box displaying the note is dependent on its parent component; if the entry component window closes, then the Note box also closes. If the note has not been saved since it was created or last altered, a dialog box will appear prompting the user to save the note before the entry window is closed. Notes are attached at the level of the component as a whole. Only one note may exist for any given component. Notes are saved when a session is saved.

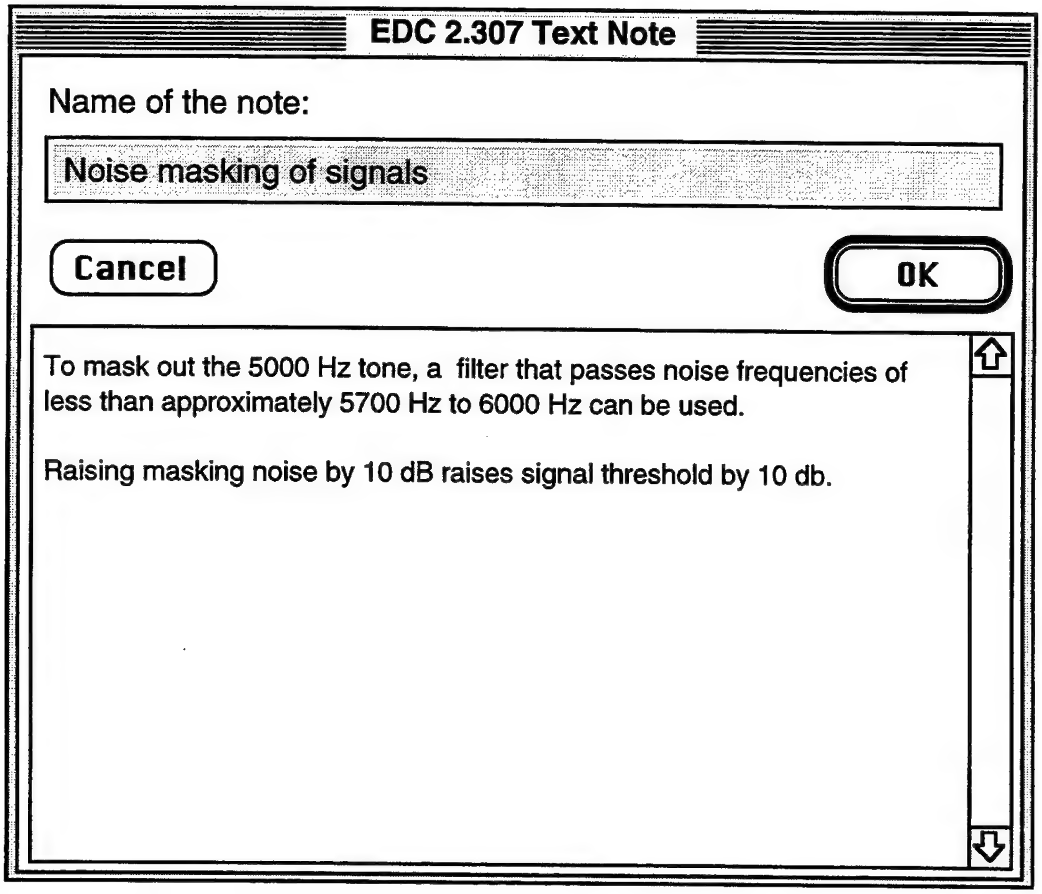


Figure 13. Dialog box for creating a user note

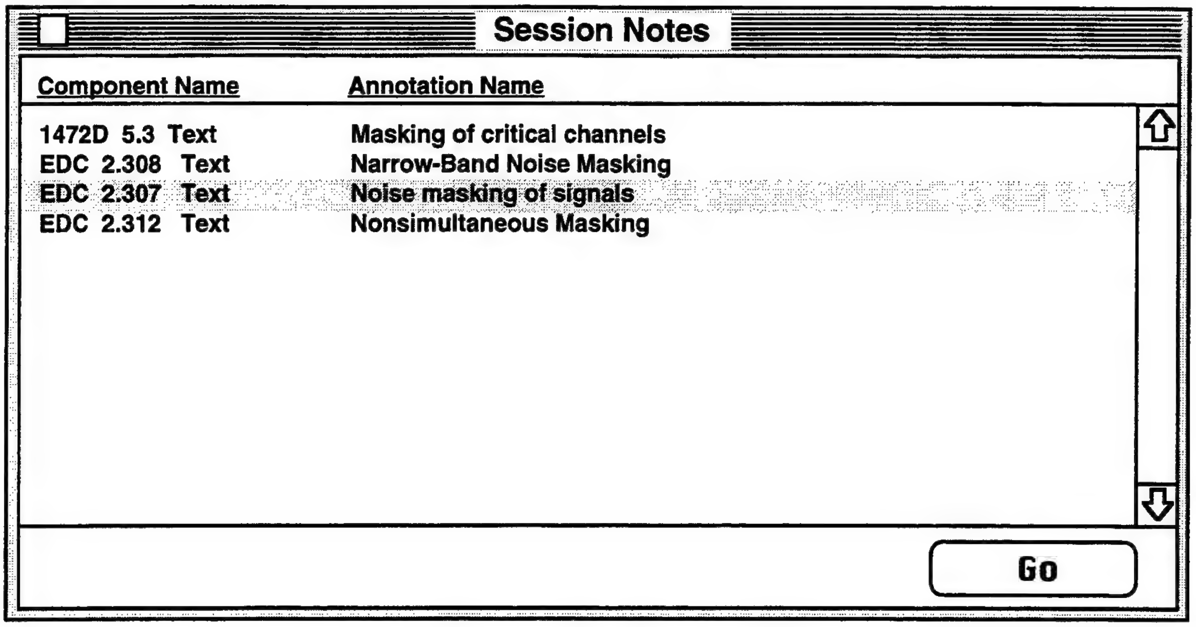
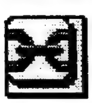


Figure 14. Session Notes window showing all components with attached notes

5.1.3 Hyperlinks

The Links feature allows users to define new relationships between components by establishing a hyperlink connection between any two components in the reference documents. If two components are linked together, then it is easy for users to navigate between these two components. Although the database contains many built-in hyperlinks to aid navigation, these preset hyperlinks represent the views of the author or editor about which components are related. The Links feature enables users to create connections between components that they themselves judge to be related to one another.



Hyperlinks for a given component are created or followed using the Links button in the Annotation region of the component viewer sidebar. If the current component contains a hyperlink, the center link of the chain will be displayed on the Links button. Clicking the Links button opens a dialog box that shows the destination components for all the previously defined hyperlinks originating from the current component (Fig. 15).

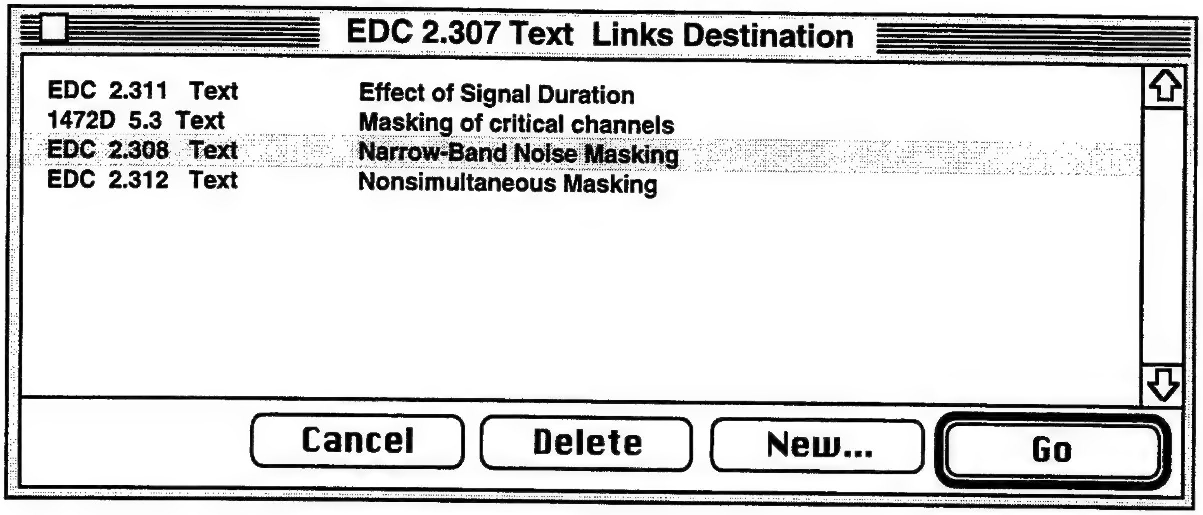


Figure 15. Link destination dialog box showing the destination components for all the hyperlinks originating from the current component

Hyperlinks are listed in the order in which they were created and show the entry number and component type of the entry component that is the link destination as well as the link name. An existing hyperlink can be traversed by selecting it from the Link dialog box and clicking Go. The destination component will be opened.

Clicking New in the Links dialog box opens a link utility window that allows a new link to be created and named (Fig. 16). The current component is the source (or origin) of the hyperlink. The user navigates to the desired destination component, then clicks Set Link in the utility window. This creates the hyperlink and closes the link utility window.

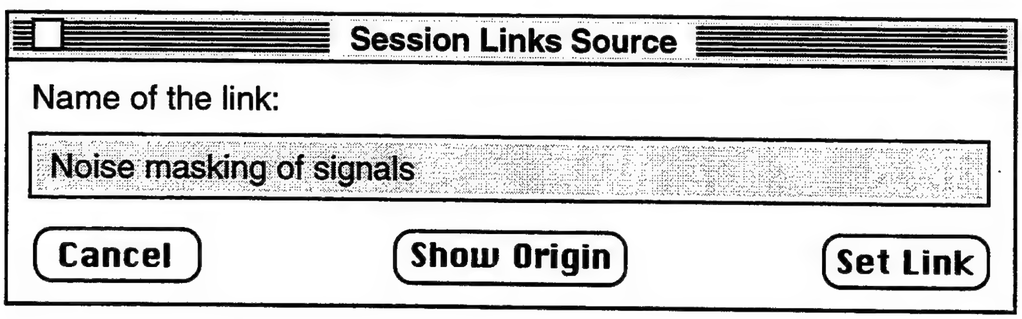


Figure 16. Link utility window for creating a new hyperlink

Hyperlinks are unidirectional from the current component to another component. That is, adding a hyperlink from component A to component B does not also create a hyperlink from component B to component A. A separate hyperlink from B to A must be created. The Link dialog box displays only the hyperlinks *from* the current component. It does not display hyperlinks *to* the component.

Selecting Links from the Annotations menu in the menu bar opens a window (Fig. 17) listing all user-defined hyperlinks for the current session. The listing shows the components that are the source points for the hyperlinks, in order by entry number, as well as the names given to the hyperlinks by the user. Users can navigate to any link origin component by double-clicking the desired component or selecting the component and clicking Go. The appropriate link origin component will be opened. The hyperlink can be followed to the destination component using the Links button in the component viewer.

Hyperlinks are attached at the level of the component (text, figure, or table). Any number of user-defined hyperlinks may be created for a given component. User hyperlinks are saved when the session is saved.

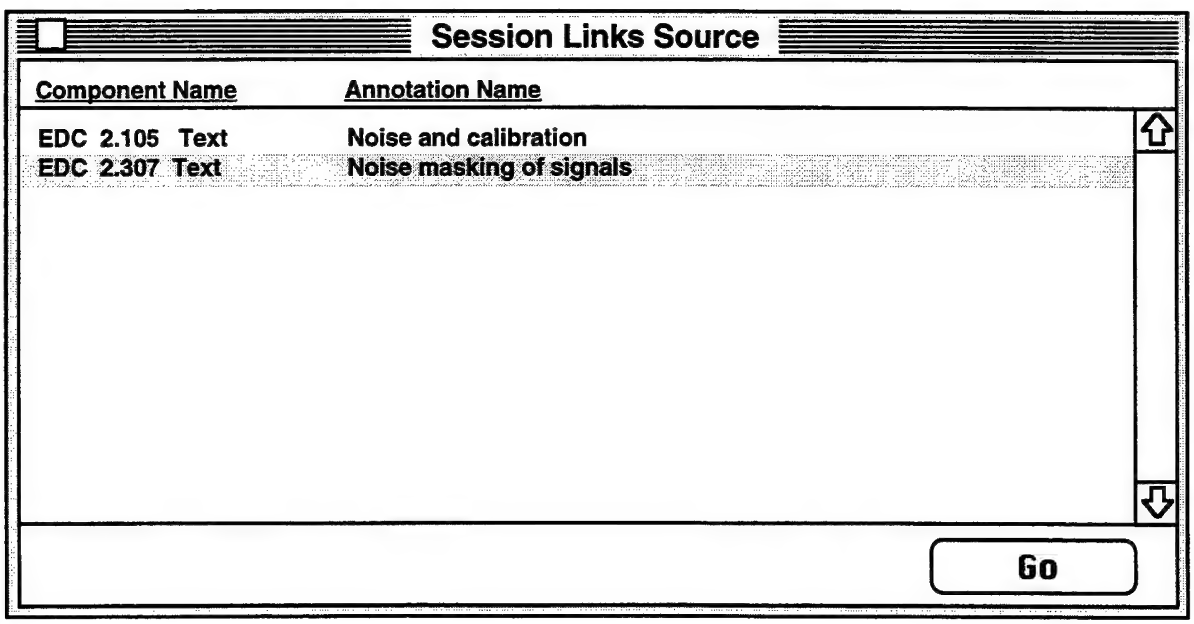


Figure 17. Session Links Source window showing all components that are the origin for hyperlinks

5.2 Session Support

Another way in which users can customize their interactions with the CASHE database is by saving the work context. Whenever a user is running the CASHE system, it is considered to be a *session*. Users can save the current session at any time and reload it later. Saving a session preserves the user's notes, hyperlinks, and bookmarks from the session, as well as the history list, and records the state of the desktop (size, position, and contents of all open component viewer windows). This same configuration is then re-created when the user reloads the session.

A new, untitled session begins each time the user starts CASHE. The user may name the current session and save it to a file. Although only one session can be active at once, the user can rename the current session, begin a new session, or reload a previously stored session in place of the current session at any time.

Saving a CASHE session allows users to continue unfinished work without disruption or to reinstate an especially useful work context. Users who are working on several projects at once can maintain a separate session file for each project focusing on a different set of database entries. Saved sessions also provide a convenient means for sharing project-related annotations or especially useful data configurations with colleagues who also use the CASHE CD-ROM.

5.3 Export and Import of Information

All of *MIL-STD-1472D* and the text of the *EDC* are in the public domain. Users of *MIL-STD-1472D* commonly extract relevant paragraphs from the standard and insert them into specifications and other contract documents. Users of the *EDC* might well wish to include portions of the information it contains in reviews and reports. Thus, when users locate interesting and useful material in the database, the system supports them in extracting this information for use outside the CASHE software.

CASHE provides for such data export through the printing, saving, and copying functions. Users can print out most text, table, or figure components of the database using the standard Macintosh print command in the File menu. When components are printed, the full bibliographic citation for the source document (*EDC* or *MIL-STD-1472D*) and the entry number and title are included to provide context.

The contents of most text, figure, and table components can also be copied to the Clipboard to be pasted into other documents or applications, or saved to a file. Text is saved in rich text format, which preserves some formatting attributes, such as boldface and italics. Tables are saved as tabbed text for convenient insertion into spreadsheets and word-processors. Figures are saved as PICT files, a format readable by virtually any Macintosh application with graphics-handling capabilities. Context information (document name, entry number and title, etc.) is included when components are copied or saved to a file.

Although the text of the *EDC* is not under copyright, some of the figures and tables it contains originally came from other sources and are copyrighted. To alert users to possible copyright infringement in exporting such items for some uses, permissions credit lines are appended whenever copyrighted figures or tables are printed or saved to a file. In a very few cases, printing and/or copying is blocked for an individual figure or table at the request of the copyright holder.

In addition to exporting material from the database documents, CASHE also allows some other types of information to be exported or imported. The DataDigitizer, the CASHE application for digitizing graphics (described in sect. 6.3), allows users both to export and import data. The data table containing the X,Y coordinates of the digitized data points can be saved to a tabbed text file for export to data-analysis applications such as spreadsheets or plotting programs. The DataDigitizer also allows users to import any PICT-format graphic for digitizing.

CASHE version 1.0 also provides some limited ability for users to import their own designs into the CASHE test benches. The Display Vibration Test Bench allows users to import visual display graphics and subject them to simulated vibration of various types and amplitudes.

6. SPECIAL USER AIDS

The target users of CASHE are design engineers and others involved in the development of human-operated systems. These users will come from a variety of backgrounds, but most will have little or no training in the scientific research areas that are represented in the CASHE database.

Computerization of the database has made it possible to develop more sophisticated tools for explicating and enhancing the perceptual and performance information in the database. Some of these tools are improvements on facilities that existed in the hard-copy version of the *EDC*. Others are entirely new enhancements made possible by the power and flexibility of the new medium. In this chapter, we describe several user aids that help users locate and understand the information presented in the database and operate the CASHE software. In the next chapter, we discuss the more sophisticated visualization tools that are also included in CASHE.

6.1 Glossary

The *EDC* is written in as nontechnical a style as possible. Because of its specialized subject matter, however, it still contains terms that may be unfamiliar to some users. To help these users understand the subject matter, the printed *EDC* provided a Glossary that defines over 500 technical terms used in the document. This Glossary is reproduced on the CD-ROM, but it is integrated with the text through hypertext linking to make understanding the material as effortless as possible. Terms defined in the Glossary are bolded when they occur in the entry text and are hyperlinked to the Glossary. Clicking the bolded term opens a Glossary window scrolled to the term's definition.

Users can also access the Glossary through the Documents menu on the menu bar. Selecting Glossary from the menu opens a two-level outline.

Level 1 is the letters of the alphabet. Level 2 is the alphabetized list of Glossary terms for a given letter. Selecting a Glossary word or phrase will open a window scrolled to the definition of that Glossary item. Once the user is within the Glossary window, other Glossary terms may be viewed by scrolling.

The Glossary is enhanced with hyperlinked items, such as cross references to related Glossary terms, cross references to *EDC* entries, or special demonstrations—all of which offer additional information about the Glossary item. Selecting a “See also” Glossary term will jump the user directly to the new term. Clicking an entry cross reference will navigate the user to an entry that supplies additional information about the concept. Selecting a demonstration hyperlink will open a window from which users can launch an audiovisual demonstration that provides a further explanation or example of the Glossary term.

6.2 Design Checklist

The Design Checklist is a special access tool that helps users identify and locate human factors data in the *EDC*. Although the Checklist also appeared in the printed version of this document, it has been expanded for the CD-ROM to provide complete coverage of all the *EDC* entries. The Checklist consists of a set of human performance questions similar to those an engineer might ask when designing control and display equipment for human operation. (Sample Checklist questions are shown in Figure 18.) The questions are sorted into categories keyed to a hierarchy of equipment-related factors. Each question is indexed, in turn, to specific entries within the *EDC* that provide information to answer the question raised. The classification scheme in the Design Checklist complements the "scientific" taxonomy of the Table of Contents with a more "design-centered" view, as opposed to the human-oriented perspective around which the *EDC* is structured.

The Design Checklist is accessed by a multilevel outline. The user can browse the outline by expanding and collapsing topic headings to find the

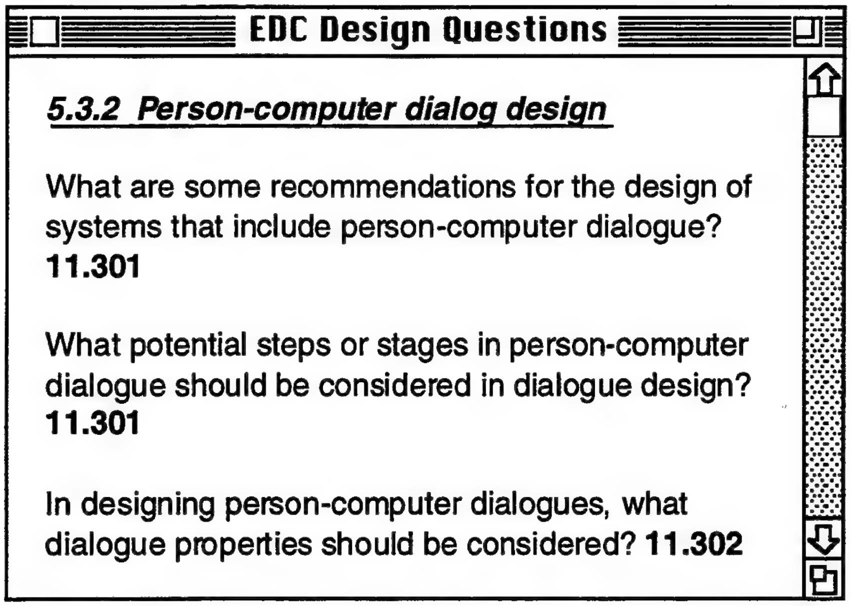


Figure 18. Sample questions from the Design Checklist (the bold numbers are hyperlinks to the entry that answers the question)

topic of interest. Clicking the desired outline element then opens the appropriate section of Checklist questions. Once in the Checklist window, the user can scroll through the selected category of questions and browse other question areas. If the user finds a Checklist question that seems relevant to the current design problem, he or she may jump to the *EDC* entry that provides information to answer that specific design question by selecting the entry reference number at the end of the question.

6.3 DataDigitizer

The DataDigitizer gives users the capability to quantify and compare the information in data graphs in order to make it more pertinent and applicable to their design interests. This functionality is intended to help users understand and visualize behavioral phenomena in a more analytical manner.

The DataDigitizer is a separate application program that can be launched from within a FigureViewer window by clicking the DataDigitizer

button on the left sidebar of the window. This opens a DataDigitizer window that contains the graph currently displayed in the FigureViewer (see Fig. 19). The graph is brought into the DataDigitizer in its current panel and overlay configuration. If a different figure panel or overlay set is desired, the appropriate configuration must be set up within the FigureViewer before the DataDigitizer button is clicked.

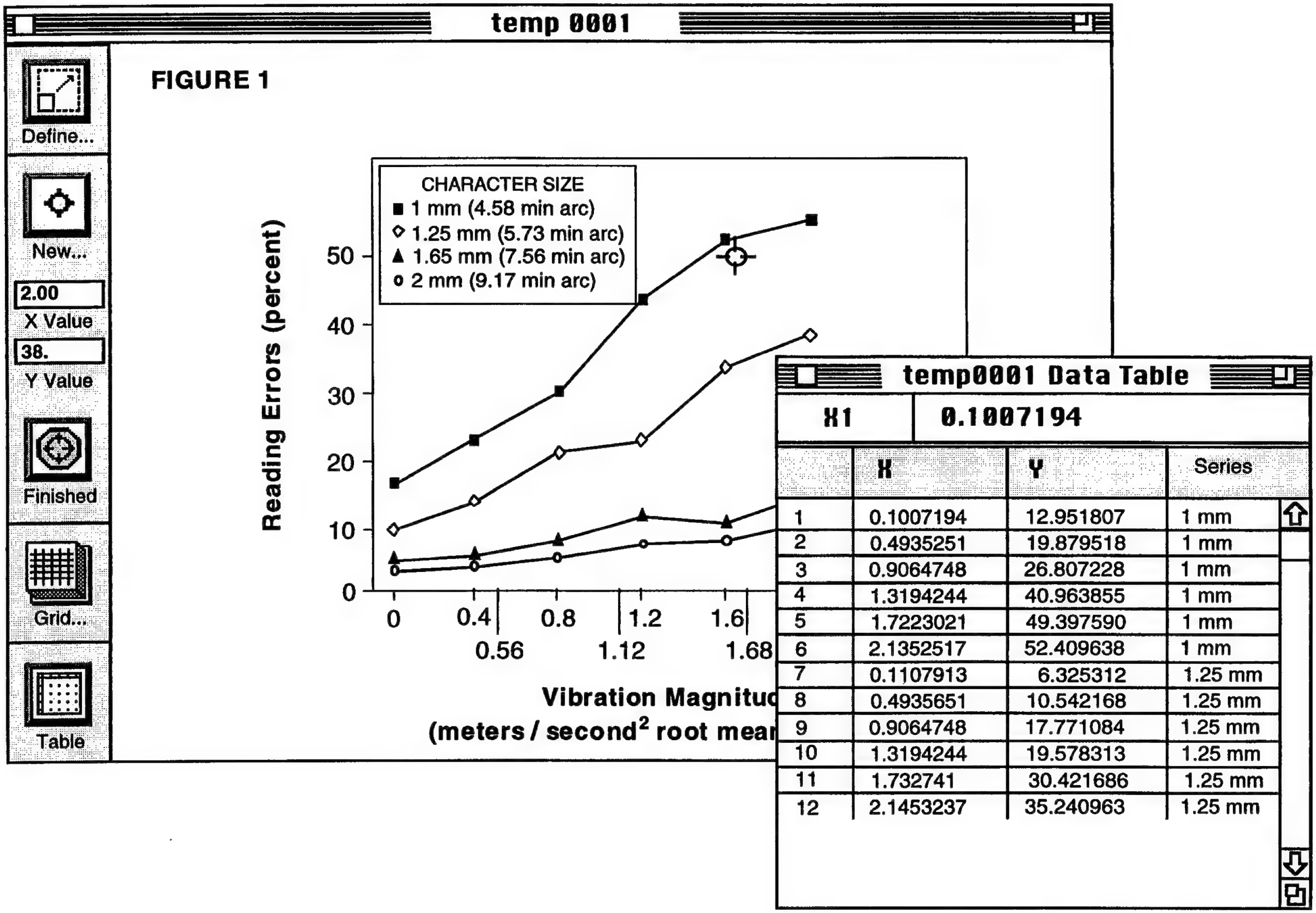
When utilizing the DataDigitizer, the user may:

- Digitize any number of points on the graph or other figure contained in the content region.
- Edit the table of X,Y coordinate values representing the digitized points.
- Save the table of coordinates to a file.
- Import any PICT file into the content region of the DataDigitizer window.
- Print the table of coordinate values.

To digitize a data graph, the user first specifies the scale factors for the graph (i.e., the relationship between screen pixels and graph units). Scale factors for the X and Y axes are specified by defining a rectangular region within the area to be digitized and then designating the exact X and Y values of the lower left and upper right corners of this region. The user also selects a scale type (linear or logarithmic) for each axis.

Once the scale factors have been defined, the user digitizes points on the graph by positioning a reticle-shaped cursor over each desired point. As the reticle is moved over the graph, the scaled X and Y coordinate values of the current cursor position are shown in readout boxes on the DataDigitizer sidebar. Clicking the mouse button records the coordinates of the current cursor location and stores them in a data table. A grid can be superimposed on the graph to aid digitization.

Multiple data series can be acquired from the graph to allow more flexible data analysis. For example, each of several curves on a graph can be placed in a different series and given a different series name, so the user can



	X	Y	Series
1	0.1007194	12.951807	1 mm
2	0.4935251	19.879518	1 mm
3	0.9064748	26.807228	1 mm
4	1.3194244	40.963855	1 mm
5	1.7223021	49.397590	1 mm
6	2.1352517	52.409638	1 mm
7	0.1107913	6.325312	1.25 mm
8	0.4935651	10.542168	1.25 mm
9	0.9064748	17.771084	1.25 mm
10	1.3194244	19.578313	1.25 mm
11	1.732741	30.421686	1.25 mm
12	2.1453237	35.240963	1.25 mm

Figure 19. Sample DataDigitizer window with open Data Table window

quantitatively compare the different curves associated with various parameter values.

The digitized data points in the table of coordinate values can be examined and edited by the user (see Fig. 19). The data table can be printed or saved to a file. The table is saved in tabbed text format so it can easily be imported into a spreadsheet or other data analysis package for further quantitative or graphical analysis.

More than one DataDigitizer window may be open at a time (though only one can be active), and several DataDigitizer windows may be associated with the figures for a single entry.

The DataDigitizer may also be accessed from the system desktop. Since DataDigitizer is a stand-alone application, it can be launched by double-clicking its icon in the CASHE folder installed on the hard disk. This opens a blank DataDigitizer window into which the user may import or paste any PICT figure to be digitized.

Figure A-7 in Appendix A summarizes the control flow for functions and commands available in the DataDigitizer window.

6.4 On-Line Help

CASHE version 1.0 provides two types of on-line help for users who need assistance in operating the product: context-sensitive help and an on-line version of the *User's Guide*.

6.4.1 Context-Sensitive Help

CASHE provides context-sensitive help by implementing the Balloon Help feature of the Macintosh operating system. Balloon Help provides on-screen commentary on the various interface elements. Users can turn Balloon Help on and off as desired. When Balloon Help is on, moving the cursor pointer to an item in the interface, such as a button or icon, will bring up a box (shaped

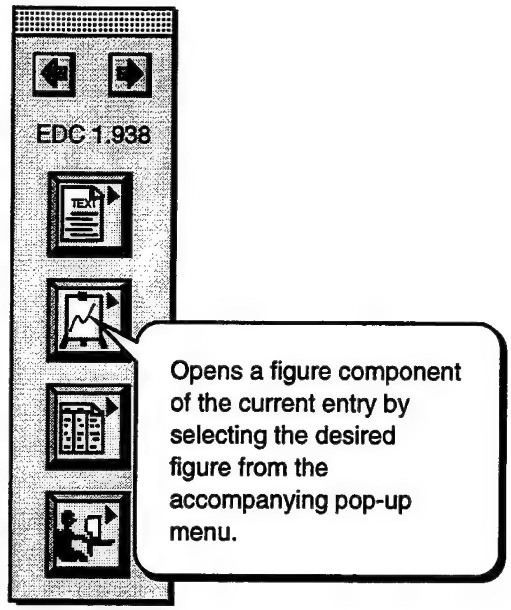


Figure 20. Example of Balloon Help

like a cartoon speech balloon) containing a description of what the item is or how it operates (Fig. 20). In this way, users get context-sensitive, task-oriented assistance in operating the CASHE interface exactly when they need it.

6.4.2 On-Line User's Guide

A comprehensive *User's Guide* is shipped with the CASHE CD-ROM (see sect. 11.1.2). An electronic version of the *User's Guide* is available on-line. The on-line guide is a separate SuperCard application that reproduces all the text from the printed version, but does not include graphics. Users access help topics through a browsing outline with expandable/collapsible headings that operates similarly to the browsing outlines in the CASHE database itself. Providing the *User's Guide* on-line makes it easy for users to access help on product operation and functionality without leaving the desktop environment.

7. VISUALIZATION ENHANCEMENTS

The CASHE electronic database improves the accessibility of ergonomics data by making available to users a large quantity of high-quality research findings in many areas of human perception and performance. However, it is also the aim of CASHE to improve the interpretability and applicability of this information by finding new ways to communicate the meaning of behavioral data to designers who have no background in the scientific disciplines from which these data were drawn.

The approaches we developed to help bridge this cross-disciplinary communication gap can be defined broadly as visualization, in that they provide a means of exploring complex human behavioral data to facilitate and enhance its comprehension. The visualization enhancements in CASHE enable users to reproduce or simulate the perceptual and performance effects embodied in the data by launching demonstrations and running mini-experiments. By directly experiencing a given behavioral phenomenon and manipulating some of the most important variables influencing it, users are able to build an intuitive understanding of the meaning of the research data and better grasp the significance of the data for current design issues. Such direct experience is especially important in applied settings like the design floor, where users are unfamiliar with the underlying concepts, methodology, and behavioral phenomena represented in the information base.

In this chapter, we describe two types of visualization enhancements included in CASHE version 1.0: (1) play-only audio/visual demonstrations, and (2) the Perception and Performance Prototyper (P^3) test benches. Because the test benches are unique to CASHE and, we believe, represent a powerful and innovative tool for enhancing the understanding of behavioral research data, we provide a detailed discussion of the issues we dealt with in designing and developing the test benches, and a full description of their general operation. (A

specific description of each individual test bench can be found in the CASHE *User's Guide*.)

7.1 Play-Only Demonstrations

The CASHE database documents contain a number of tutorial figures and tables that illustrate stimulus configurations, explain behavioral concepts, or illustrate perceptual phenomena to help users understand and interpret the human perceptual and performance data in the database. The electronic database takes advantage of the capabilities of the computer environment to enhance many of these figures with animations or audio clips that allow users to directly experience the effects described. The database contains 71 of such play-only demonstrations. Table 1 shows the general subject areas in which play-only demonstrations are provided.

These play-only demonstrations are "canned" presentations with preset stimulus values designed to produce specific effects. They cannot be altered by the user. The simplest demonstrations reproduce a stimulus or

Table 1. General Subject Areas of the 71 Play-Only Demonstrations

Color vision phenomena	Sound stimulus types
Size illusions	Decibel scale
Shape/slant perception	Harmonic and intermodulation distortion
Motion perception	Auditory masking
Induced motion	Auditory temporal resolution
Apparent motion	Loudness phenomena
Kinetic depth effects and	Monaural vs binaural sound
Kinetic motion	Interaural effects
Anorthoscopic perception	Auditory grouping phenomena
Figural aftereffects	Speech level
Contingent aftereffects	Peak clipping of speech
	Linguistic samples

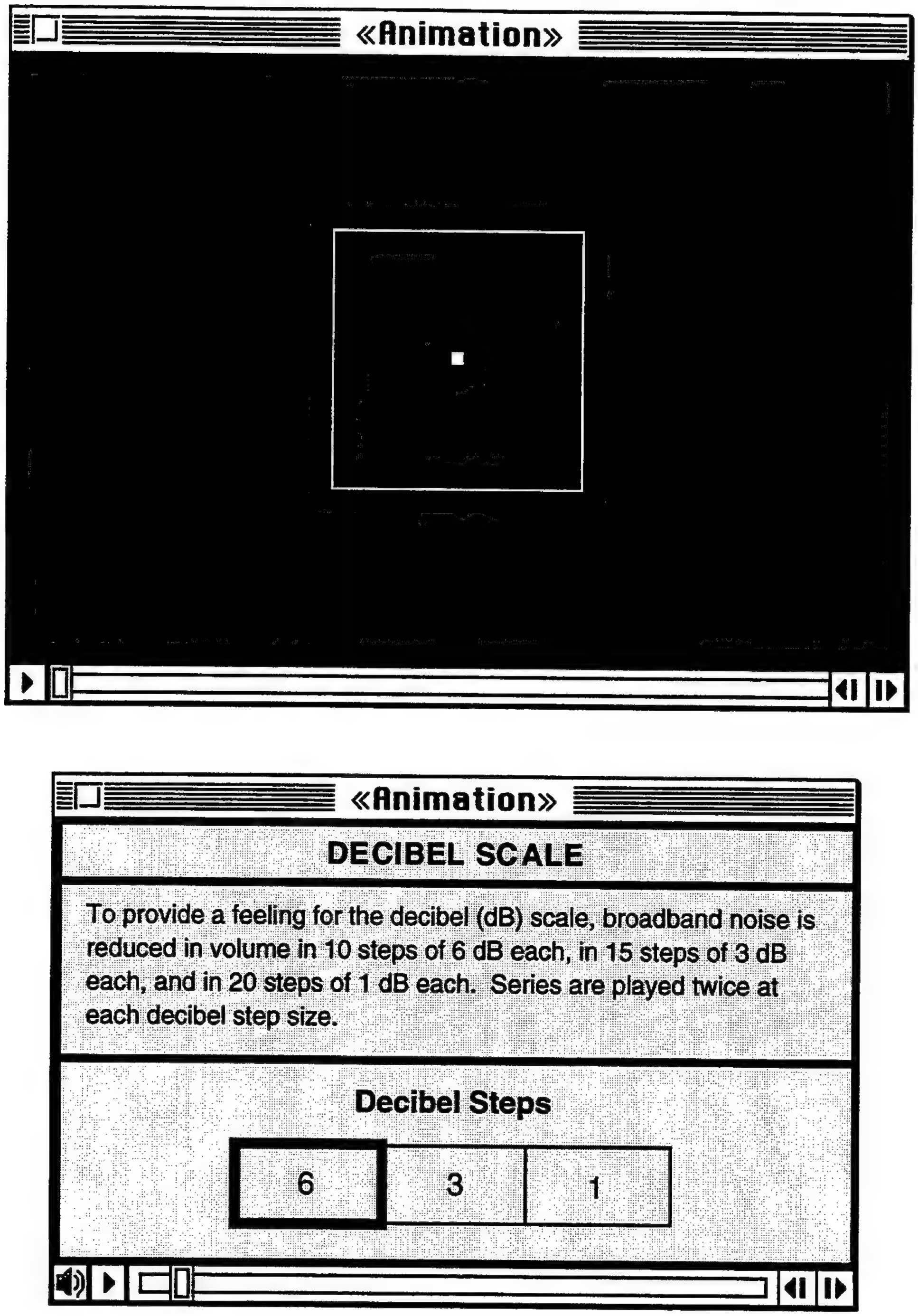


Figure 21. Sample screens from play-only demonstrations. The visual animation shown at top allows the user to experience induced motion. When the animation is run, the outline square moves left, which causes the stationary white square to appear to move right. The auditory demonstration shown at bottom plays noise samples whose intensity is reduced in measured steps. The heavy box on the readout at bottom indicates the decibel step size currently being played.

illustrate a concept to help users better understand the data in the database and the experimental conditions under which they were collected. For example, users can hear samples of white noise or amplitude-modulated tones or view an animation that illustrates the difference between object-relative and subject-relative motion. Other demonstrations allow users to experience perceptual phenomena such as induced motion, kinetic depth effects, and auditory grouping phenomena. There are also more complex play-only demonstrations, for example, timed stimulus sequences that allow users to experience figural or contingent aftereffects and descending-staircase demonstrations of signals in noise that let users make simple measurements of auditory masking effects.

All play-only demonstrations are linked to database graphics. Figures and tables with available demonstrations are marked with an asterisk in entry component menus and browsing outlines. Users access the demonstrations by clicking a "movie" or "loudspeaker" icon in the figure or a hyperlink indicator in the table. The demonstrations are presented using the Apple QuickTime MoviePlayer. The MoviePlayer window contains Start, Stop, Rewind, Frame Forward, and Frame Back buttons that enable the user to play and control the demonstrations. Figure 21 shows typical displays for visual and auditory play-only demonstrations.

7.2 Perception and Performance Prototyper Test Benches

The centerpiece of CASHE visualization is the set of test benches known collectively as the Perception and Performance Prototyper (P^3). The test benches are interactive software applications that link to and amplify the information in topically related groups of entries in the CASHE database. Test benches help illustrate and further explain a perceptual or performance effect described in the entries by reproducing or simulating the effect so users can experience it firsthand. Unlike the play-only demonstrations, the test benches are controlled by the user and can be customized to address specific needs and interests. The test benches let users set up visual targets or auditory stimuli,

adjust important stimulus characteristics, and manipulate presentation and task conditions to explore and analyze the most significant factors governing a given behavioral phenomenon. Users can experiment with combinations of variables that are directly related to their immediate design issues and "prototype" simple design solutions.

We struggled with a number of important conceptual issues in determining how to construct test benches that could support users in interpreting and applying the perception and performance data in the CASHE database. These included: (1) what could be simulated and how; (2) what the scope of each test bench should be; (3) how faithfully the test benches should (or could) mirror the original research studies; and (4) how to structure the test benches to support different levels of use and user expertise. In the following sections, we will first discuss these theoretical issues and how we addressed them, and then describe the design and operation of the test benches.

7.2.1 Approaches for Simulating Perceptual and Performance Data

Our first task in developing the test benches was to determine which behavioral effects could be simulated successfully and what methods could be used to reproduce the effects on a desktop display.

The documents in the CASHE reference database contain over 1100 entries covering complex and varied perceptual and performance effects, phenomena, and relationships. When we reviewed these entries to assess how to represent the information in interactive test benches, it quickly became apparent that no single approach was adequate for simulating all these effects. Some behavioral phenomena are robust and can be well accommodated on computer displays. Other phenomena are "statistical" effects that become visible only when large blocks of performance trials are analyzed. Finally, some phenomena, such as threshold effects, simply cannot be reproduced on the Macintosh platform because of software/hardware limitations.

We developed four different approaches to deal with these different classes of information and maximize the value of the test benches: (1) reproduction of effects, (2) simulation of effects, (3) self-testing, and (4) data access.

Reproduction of effects: For many information items in the database, it is possible to re-create the relevant phenomenon or effect on the computer so the user can experience the effect firsthand. For example, there are test benches allowing users to observe how changes in target brightness and contrast affect visual acuity (visual resolution), how the location of a target in the visual field influences the perceived flicker of the target, how the frequency of a sound signal affects its audibility, and how different target and display characteristics influence the speed and accuracy of visual search.

Simulation of effects: Some perceptual and performance phenomena cannot be reproduced directly but can be simulated effectively to provide a feeling for the phenomena and enhance understanding. For example, although the computer monitor cannot be physically vibrated, mathematical modeling makes it possible to simulate certain types of display vibration by moving a visual image on screen or by blurring the image to approximate the retinal smear that would result from instability of the display in an actual vibration environment. Such simulation allows the user to make useful qualitative judgments regarding display legibility under different vibration conditions.

Self-testing: Some database entries treat subtle performance effects that become visible only through a formal analysis of performance data obtained under different stimulus conditions. To simulate these effects, some test benches allow users to run mini-experiments in which user data are collected. Users are guided through a series of structured trials in which stimulus characteristics or presentation conditions are varied systematically and user performance data are collected and analyzed. A graph of the user's results is then presented showing how performance varies for different defined conditions. For example, in the Visual Search Test Bench, users perform blocks of trials in which they must locate a specified target in a

display of many objects. Response speed and accuracy are measured and graphed for the user to reveal how various factors such as target coding, target-background contrast, and display density affect search performance.

Data access: Some perceptual and performance effects (such as threshold-level phenomena) cannot be reproduced or simulated using standard computer displays in office environments. Several large blocks of entries contain information of this type. Although these effects cannot be reproduced for direct experience, it was felt that the data could be made significantly more usable for designers if they could be represented in a test-bench-type interface that would help users identify the data germane to specific design problems. A special data-finder category of test bench module was developed to achieve this goal. Users employ a control panel to specify the stimulus conditions in which they are interested. The data finder then displays research data from the database that match these conditions. In the Sound Localization Test Bench, for example, users specify auditory signal type and signal frequency, and are then shown data indicating how accurately listeners can localize the indicated signal. Users can also designate the performance measure desired: minimum audible angle (smallest discriminable difference in the angular location of two sound sources) or localization error (angular error in identifying the position of a sound source). Users can obtain details of the experimental study that collected the data by jumping to the appropriate entry in the database via cross-reference links provided in the test bench.

Depending on the type of performance data encompassed, some test benches rely on only one of the approaches described above, while others use a combination of approaches to accommodate difference aspects of the available data.

7.2.2 The Scope of the Test Benches

In addition to determining the general approaches for simulating perception and performance data, we also had to define the general scope of individual test benches. The issue here was one of universality versus

customization. At one extreme, we could create a small number of "generic" test benches, each supporting a factorial mix of stimuli, presentation conditions, and task types to allow users to reproduce virtually any set of conditions described in the database. At the other extreme, we could provide an extensive set of unique, highly specific test benches, each customized for a different entry in the database. Neither of these approaches was considered optimal or feasible. A broad, generic test bench would burden the user with lengthy set-up routines and a myriad of choices that might be confusing and intimidating to a user inexperienced in the subject matter. A large set of narrowly focused, completely customized test benches would fragment and compartmentalize the information, making it more difficult for users to see unifying relationships among different database entries. In addition, the programming complexity of the first approach and the scale of the second made both too costly and time consuming for practical implementation.

Instead, we adopted an approach that was midway between these two extremes. An information-clustering strategy was developed to allow some standardization of tasks and conditions in the test benches. Database entries treating related topics were grouped together to share a single master (standardized) test bench with a standard control panel, standard task representation, and standard interface. The advantages of such an approach were that it (1) reduced the potentially infinite number of variables (i.e., stimuli, presentation conditions, and task types) to those actually utilized in a specific set of database entries; (2) decreased the number of test benches necessary because each test bench could serve multiple entries; (3) provided a broader prototyping capability by making available the variables treated in all database entries that access the test bench and not just those for an individual entry; and (4) through this consolidation, provided a means of making users aware of additional relevant variables they may not have considered had each test bench covered only a single data item.

With this clustering strategy in mind, the database entries were surveyed to identify the major underlying behavioral concepts and select good

Table 2. P^3 Test Benches Developed for CASHE Version 1.0

Auditory Sensitivity Test Bench	Speech Intelligibility in Noise Test Bench
Display Vibration Test Bench	Visual Acuity Test Bench
Flicker Sensitivity Test Bench	Visual Optics Test Bench
Manual Control Test Bench	Visual Search Test Bench
Motion Perception Test Bench	Warnings and Alerts Test Bench
Sound Localization Test Bench	

candidates for test benches. Over 60 potential test bench topics were identified. Of these, the set of 11 test benches shown in Table 2 were selected for development for version 1.0 of CASHE (further details on the topic selection process can be found in sect. 10.1):

7.2.3 Design of the Test Benches

In designing the content and structure of each individual test bench, a major concern was how faithfully the test benches should attempt to reproduce the conditions used in collecting the data contained in the reference database. One option we considered was to create test benches that would enable users to accurately reconstruct the stimulus and task conditions of the original research studies reported in the entries, in effect providing a computer-based human performance laboratory. However, the technical limitations of the Macintosh computer environment turned out to be the determining factor here. Simulating the stimulus and task conditions of even a majority of the original studies proved impossible using current desktop computer technology. Personal computer systems allow only very limited manipulation of many critical stimulus characteristics such as luminance, size, sound frequency, and volume level. Even within this limited range, there is no means, through software alone, to calibrate visual or auditory presentations to ensure that the intended stimulus levels are actually being created. There is simply too much variability in the hardware components users may have installed in their systems. Not only is there wide disparity

among different brands, but there may also be significant variations among different models from the same manufacturer and even identical components of different ages. The environments in which the test benches will be run are also highly variable (e.g., there is no way of knowing ambient light or noise levels). Such variability makes it virtually impossible to standardize presentations to exact predetermined stimulus values.

Matching the real world in every detail has a seductive appeal in simulation design. Nevertheless, duplicating reality is not always a functional necessity, in spite of its aesthetic attractiveness. To achieve the goal of aiding users in understanding and interpreting perceptual and performance information in the database, the test benches must make it possible for users to experience and explore the effect described. However, high-fidelity reproduction of the original experimental conditions is not always necessary in order to generate the required perceptual effect for the user. The design approach we adopted was to create test benches that could provide pedagogical illustrations conveying the essence of the perceptual and performance phenomena described in the database entries. Users are able to experience these effects firsthand and investigate various aspects of the effects; but they are not necessarily able to duplicate the exact conditions of the original experimental studies. This approach was also felt to be more consistent with the way designers are likely to use the CASHE product: namely, they will operate the test benches to get a general feel for a particular perceptual or performance effect, but will consult the detailed quantitative figures and models in the reference database for specific data to support design decisions.

In keeping with the basic goal of providing a pedagogical experience of the behavioral phenomenon, we chose stimuli and conditions for each test bench so as to maximize the robustness of the effect being illustrated and to provide the types of manipulations that will help users relate their experience of the effect to design needs and issues. The test benches may include stimuli and parameters similar to those used in studies reported in the *EDC*, when

this is technically feasible. However, creating a good experience of the effect, rather than matching the *EDC* entries exactly, was our driving goal.

Most designers are trained in engineering disciplines and have little background knowledge of the behavioral effects illustrated in the test benches. Another important design concern was how to structure the test benches to make it easy for inexperienced users to investigate the effects and make meaningful examinations of the variables influencing them, while at the same time supporting more experienced users in custom-tailoring the presentations to their immediate design needs.

To address this concern, we structured each test bench as a set of self-contained modules or topics that vary in the level of user interaction supported. Standard modules are tightly structured to guide inexperienced users in focusing their explorations on important variables and performance relationships, while custom options modules provide customization features for more advanced users.

Standard modules address relatively narrow aspects of the phenomenon being presented to provide the designer with a quick experience of the basic effects. Each standard module focuses on one or two of the major variables influencing the perceptual or performance phenomenon addressed in the test bench and allows the user to manipulate these variables and observe how the effect changes. The ranges of the variables are generally constrained to a small set of values selected to maximize the experience of the effect and illustrate the major perception and performance relationships.

Custom options modules offer a full-scale control panel that lets users manipulate many more stimulus characteristics at once than the typical standard module and provides greater control of individual parameters. These modules can be used to create stimulus presentations and experimental tasks custom-tailored to the user's individual concerns.

A typical test bench includes several standard modules as well as a custom options module to support different levels of user interaction. For

example, the Visual Acuity Test Bench contains separate standard modules focusing on target brightness and contrast, visual field location, and motion that allow users to manipulate these variables and assess their effect on visual resolution. It also contains a custom options module that allows users to manipulate all these target characteristics simultaneously, provides more fine-grained control of these variables, and broadens the number of presentation characteristics that can be addressed.

Another aspect of test bench design that makes it easy for inexperienced users to interact with the test bench is the provision of context-sensitive default values. Default settings are always supplied for all parameter values, so users can enter a test bench module and launch a demonstration immediately, with no need for a lengthy set-up. Default settings are selected to provide a good experience of the phenomenon and are based on the conditions reported in the database entries where feasible. To provide continuity, when users move from a standard module into a custom options module, the custom control panel settings in the custom options module are reset to the last values used in that standard module.

7.2.4 Test Bench Operation

Each test bench is programmed as a separate application. To promote transfer of learning and ease of use, however, all test benches follow the same general structure and have a standardized interface with common functional elements and uniform graphic style. Test benches run in a TestBenchViewer (or window) with its own menu bar and functional controls. Figure 22 shows the standard TestBenchViewer for all test benches.

Utilizing the TestBenchViewer, users can:

- View a selection menu and choose a test bench topic module to run;
- Display background information about the subject covered in the test bench;

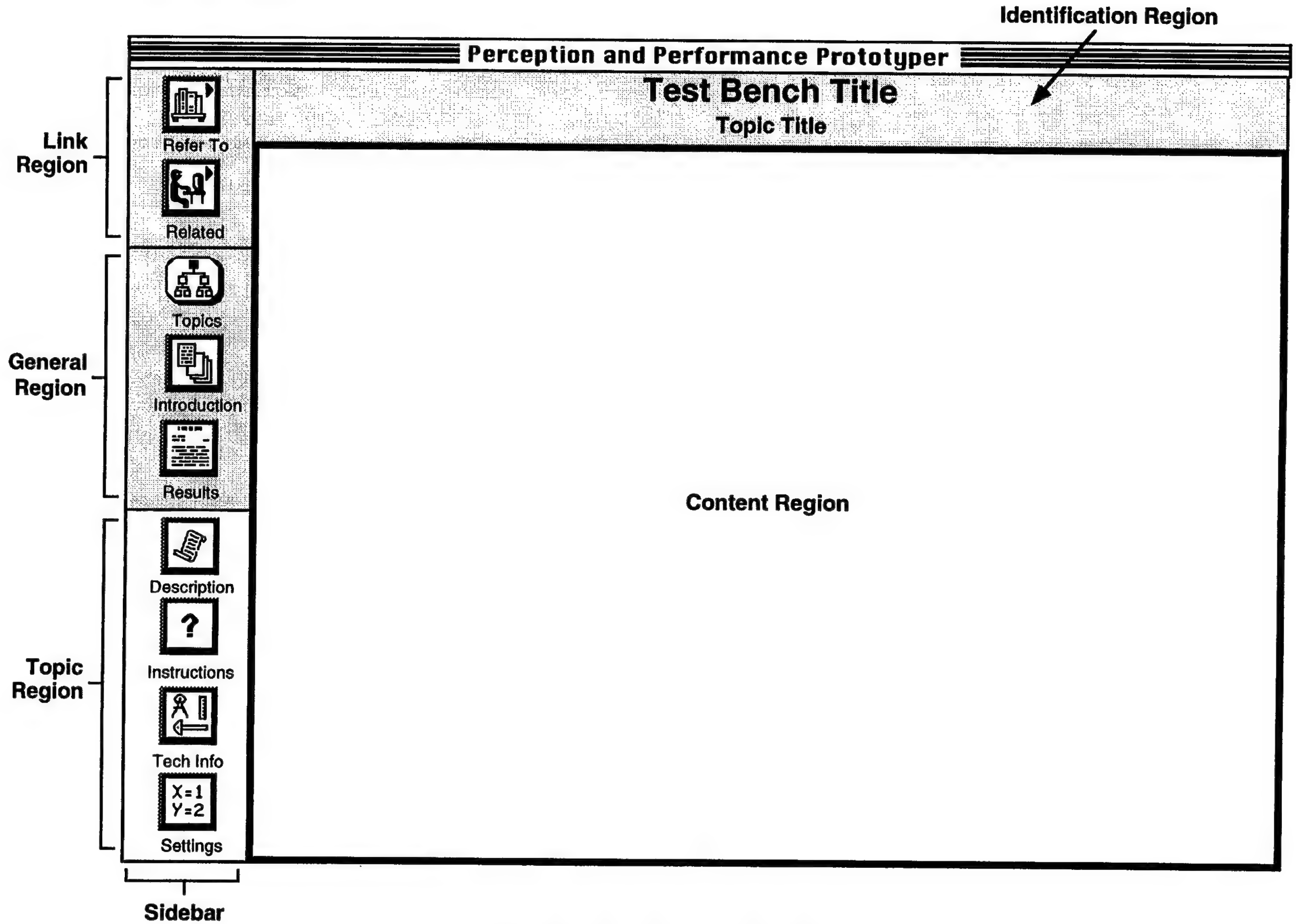


Figure 22. Standard TextBenchViewer window

- Link to database entries containing research data or design criteria related to the test bench subject as a whole or to the subject of individual test bench modules;
- Link to other test benches on related subjects;
- Access on-line commentary to help in operating the test bench, understanding the test bench topic modules, and interpreting observations and results;
- Display, edit, or print performance results or observations from the current test bench session;
- Save results and observations in a form that can be read by other applications; re-load results saved previously;
- Save customized parameter settings to a file; re-load previously saved settings to re-create a given set of experimental conditions.

The TestBenchViewer window is divided into three major areas: (1) an identification region at the top, which displays the name of the test bench as well as the name of the individual test bench module currently being viewed; (2) the sidebar, which contains buttons to invoke general functions available for all test benches; and (3) the content region, where the test bench displays are presented.

The buttons on the sidebar are grouped by type of function and level of operation into a link region, a general region, and a topic region. The buttons in the link region link users to other CASHE locations containing information related to the subject of the test bench. Users can access these buttons from the topic selection menu as well as from individual test bench modules (although their content may be different at each level). The buttons in the general region provide information about the test bench as a whole and are accessible from both the topic selection menu and the individual test bench topic modules. The buttons in the topic region are active only from within an individual test bench module and provide information related to that module. The operation of the buttons in each region is explained in detail below.

7.2.5 Topic Selection Menu

All test benches open with a topic selection menu listing the modules available for that test bench (Fig. 23). If the user has accessed the test bench from a database entry, one of the modules in the menu may be flagged with a "pointing finger" icon. The flag indicates that the topic of this module is especially closely related to that of the entry from which the test bench was accessed. Inexperienced users can begin their exploration of the test bench with this recommended topic module. A few test bench modules require specific hardware configurations, such as a color monitor or earphones. Any special hardware requirements are noted in the selection menu listing for that module.

Clicking the title of any topic module in the menu will launch that

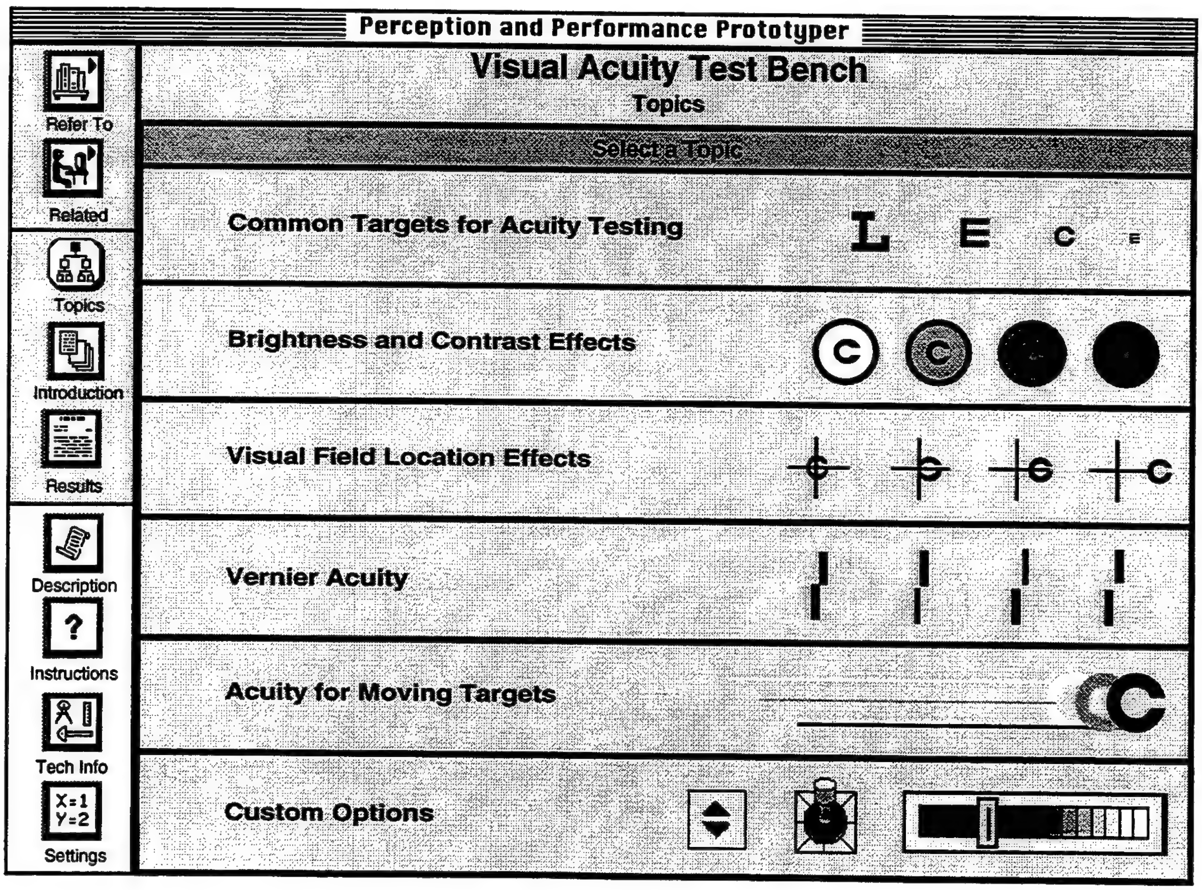


Figure 23. Sample topic selection menu

module. Users can return to the topic selection menu whenever they wish by clicking the Topics button on the sidebar.

Several higher level controls are also available to users from the topic selection menu. Clicking the Refer To button displays a pop-up menu of entries in the *EDC* or *MIL-STD 1472D* documents that provide more detailed information about the general subject of the test bench. Selecting an item from this pop-up menu opens the text component of that entry at the front of the desktop.

The Related button provides a pop-up menu of other test benches in related areas. Users can launch any test bench on the menu by selecting it from the menu (the current test bench will be closed).

Clicking the Introduction button displays an Introduction that provides background information and an overview of the subject addressed in the test bench.

The Results button opens the Results text window. This window reports parameter settings and user performance data for any mini-experiments that have been run from the test bench during the current session.

7.2.6 Standard modules

Standard topic modules allow users to set up simple demonstrations or mini-experiments designed to produce specific perceptual or performance effects. The opening screen of a typical test bench module contains several sets of software controls (radio buttons, arrow controls, sliders, etc.) for manipulating various characteristics of the stimulus or presentation conditions. Depending on the module and test bench, for example, users can adjust variables like target size and brightness, sound frequency, or control-display gain. The main screen also contains a display area that shows sample visual targets for visual presentations or readouts for following sound stimuli for auditory demonstrations as well as buttons to launch presentations or

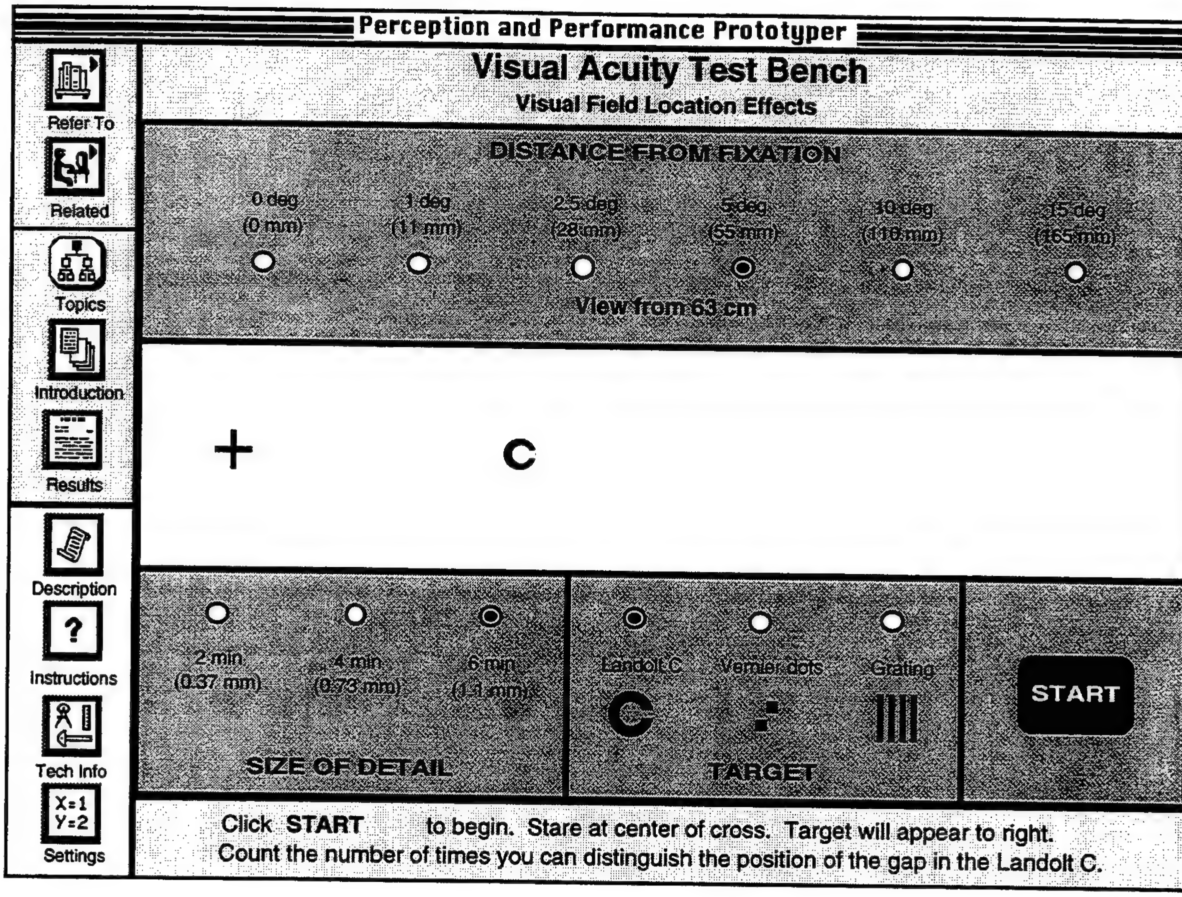


Figure 24. Main screen from a typical standard topic module

mini-experiments. (Figure 24 shows a sample opening screen from a topic module.)

The Description, Instructions, and Tech Info buttons on the sidebar provide access to text boxes containing supplementary information to help users understand and interpret the demonstration in the topic module. Clicking the Description button displays a descriptive summary that introduces the topic of the module and outlines expected effects and general findings in this topic area. Detailed instructions for operating the module can be accessed by clicking the Instructions button (although the module screens are designed to enable users to set up and operate mini-experiments with no further aid). Clicking the Tech Info button displays supplementary technical information regarding stimulus generation, performance measurement methods, and data interpretation that more advanced users may find helpful

in interpreting and generalizing the effects illustrated in the module. Users can print out the information in any of these text boxes.

The Refer To button at the top of the sidebar displays a list of entries in the *EDC* or *MIL-STD-1472D* that relate to the topic of the module. Users can open any entry of interest by selecting it from this menu list. Users can also view the general test bench Introduction or access other test benches on related topics by clicking the appropriate sidebar buttons.

Clicking the Settings button at the bottom of the sidebar opens a Settings window (Fig. 25). The Settings window lists the current value of all controls on the topic module screen as well as selected other stimulus parameters that remain constant across control settings. The purpose of the Settings display is to help users keep track of conditions that produce certain perceptual results and to make it easier for users to reproduce specific stimulus configurations and annotate their observations.

The contents of the Settings window cannot be edited, but they can be copied to the Clipboard using the Copy All command in the Edit menu. When the Settings window is active, users can print the window contents or save it to a text file using the Print, Save, and Save As commands in the File menu.

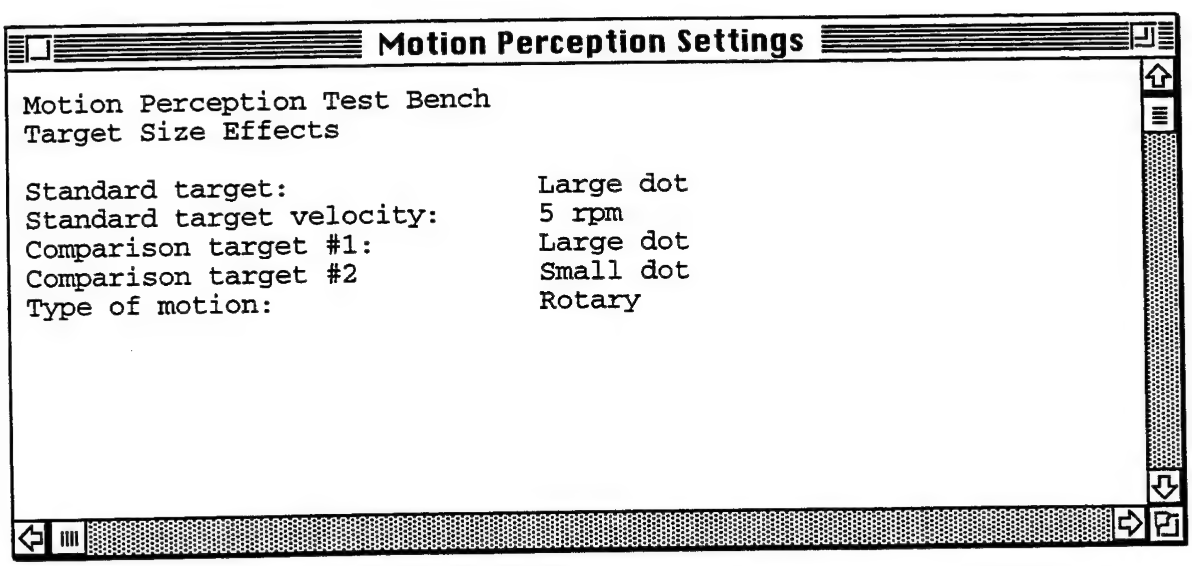


Figure 25. Typical Settings window display

Once users have set the stimulus controls on the main screen to create the desired conditions, they can initiate a visual target presentation, play an auditory demonstration, or launch a mini-experiment. Auditory presentations play from the main screen of the topic module. A changing visual indicator on the main screen helps users keep track of which signals are being played as the demonstration progresses. In modules illustrating visual phenomena, the main screen is replaced by a special target display when the demonstration is initiated. Some visual presentations are timed and return users to the main screen as soon as the presentation is complete. In others, the presentation continues until the user halts it.

In data-access modules, no demonstration is presented. Rather, users set the controls on the main screen to reflect the conditions in which they are interested, then click a button to display published research data from the *EDC* that match these conditions.

When the module supports user self-testing, one or more blocks of experimental trials will be run. After all trials are completed, the user's performance results are calculated, analyzed, and reported to the user in a Results dialog box (Fig. 26). The dialog box displays a data graph showing performance for each condition of the mini-experiment. The same data are also presented in tabular form. Users can print the contents of the Results dialog box if desired.

Users' experimental results are also recorded in text form in a Results window (Fig. 27). The Results window lists the test conditions used and reports the user's response data. The list of test conditions duplicates the table of stimulus characteristics and other parameters found in the Settings window. The user's data may be a reaction time value, accuracy data, or some other response measurement, depending on the test bench.

As users visit other modules and run additional experimental tasks, the new performance data are appended at the bottom of the Results window.

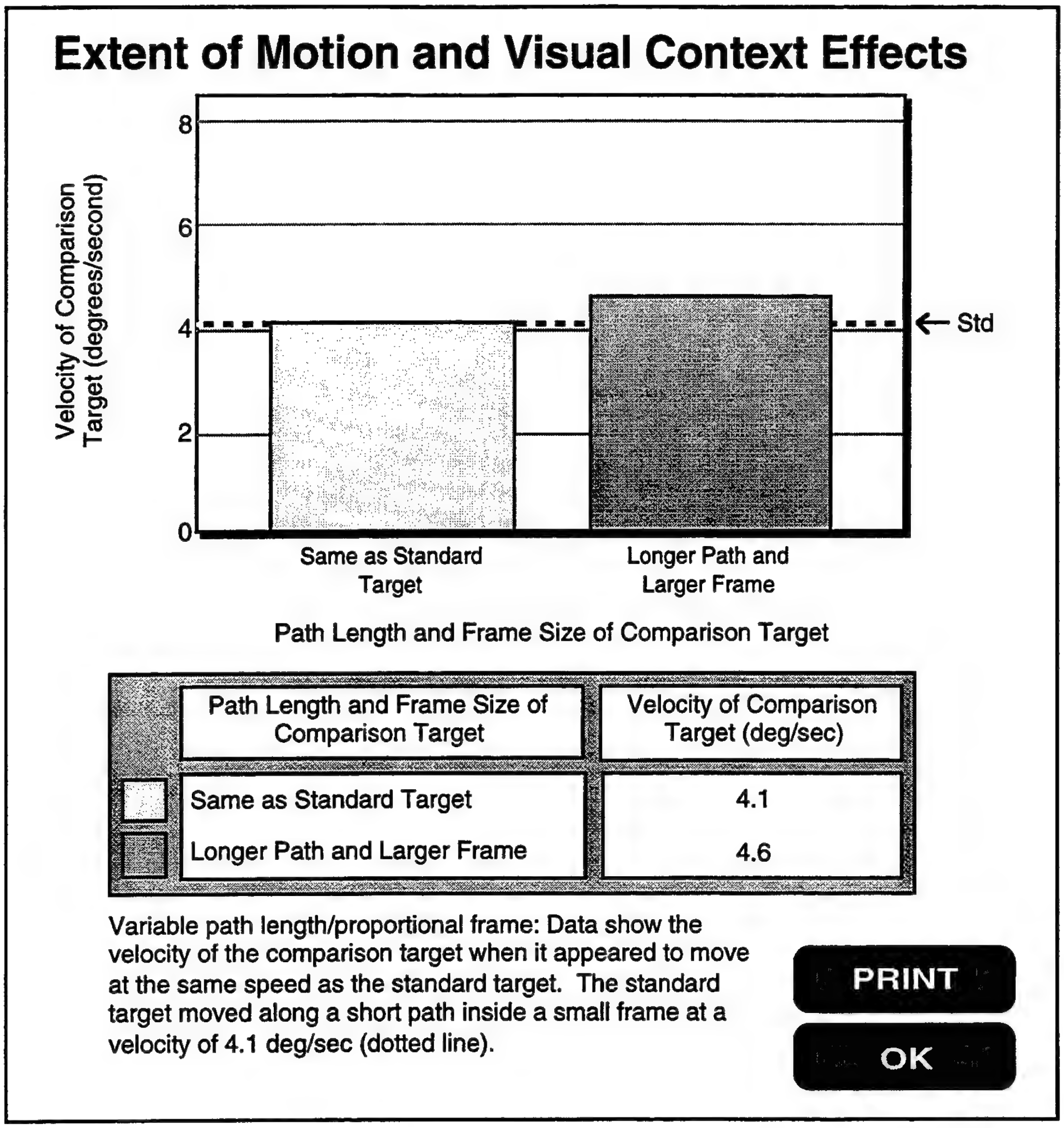


Figure 26. Typical Results dialog box display

Users can view these cumulative results at any time by clicking the Results button on the sidebar.

When the Results window is active, users can print its contents or perform simple text-editing functions on the text in the window. The window contents can also be saved to a file in ASCII format using the Save Results or Save Results As commands in the File menu. A previously saved Results file can be opened from the File menu. The file that is opened will replace the

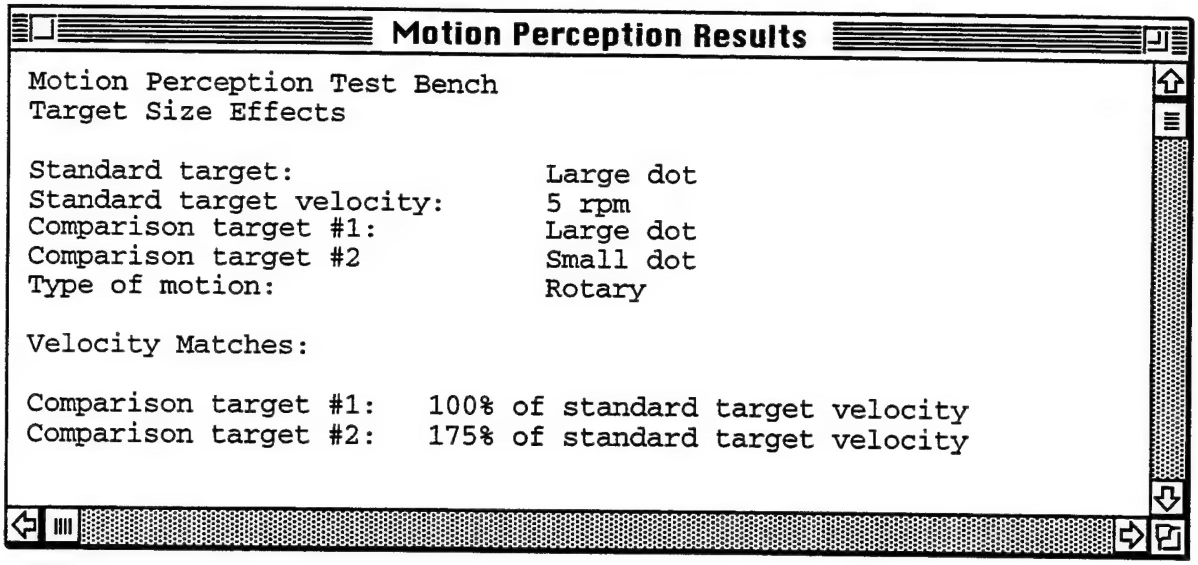


Figure 27. Typical Results window display

current contents (if any) in the Results window, and any new experimental results will be appended to this file.

In modules without user data collection, no information is written automatically to the Results window. Instead, the Results window can be used as a note pad for recording informal findings. One convenient way for users to structure their annotations is to copy the content of the Settings window into the Results note pad and then describe the perceptual effects observed with those control settings.

Designer engineers typically will consult CASHE to resolve a particular design issue or problem; thus, it was felt they might be unwilling to embark on a test bench that requires a substantial expenditure of time. To support this type of use, all test bench modules are designed so users can obtain a good feel for the effect in ten minutes or less. The on-line Instructions notify users of the approximate time required to complete all mini-experiment modules.

7.2.7 Custom Options Modules

Custom Options modules are special topic modules that allow users to create custom-tailored target presentations and experimental tasks. Although they operate similarly to standard modules, Custom Options modules allow users to manipulate many more stimulus parameters at once than the typical module screen and may also extend the operating range of some parameters. Users can also save and load specific experimental setups so that target presentations and tasks can be easily repeated.

The main screen for Custom Options modules consists entirely of a control panel through which users manipulate many different target and task characteristics (Fig. 28). When a Custom Options module is entered, the controls will automatically be set to reproduce one of the target presentations

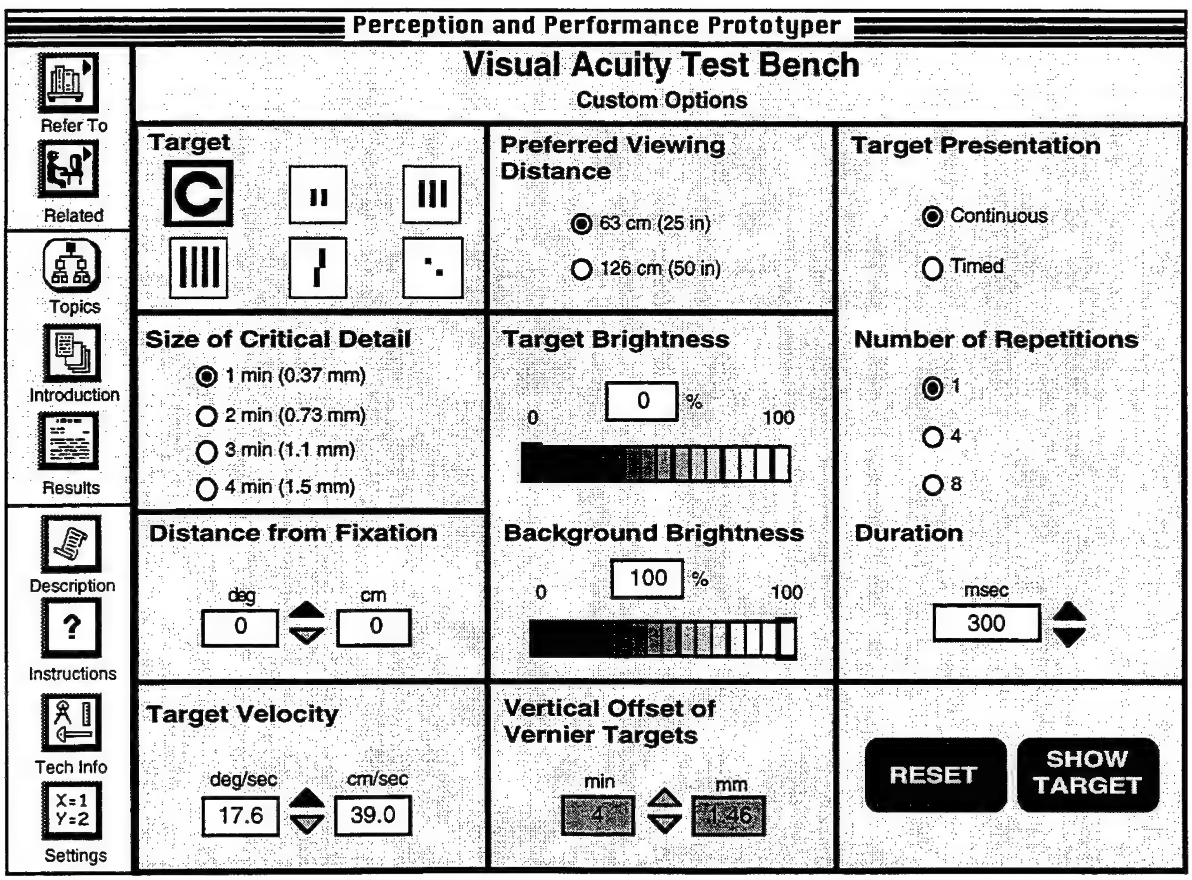


Figure 28. Typical control panel for Custom Options topic

or experimental tasks that was available in the last standard module the user visited before moving to the Custom Options module. (If the Custom Options module is entered before a standard module is visited, the parameter settings will be test bench default values).

At any point, users can click a Reset button to return the control panel settings to the parameter settings in effect when the Custom Options module was first entered. The File menu on the menu bar allows the current settings of all the controls in the Custom Options control panel to be saved. A parameter file saved previously can also be loaded using the File menu. Opening a parameter file will change the values of all controls in the control panel to the settings saved in the parameter file. Thus it is easy for users to save and later reinstate a given experimental setup.

Like standard modules, Custom Options module create visual or auditory presentations or launch mini-experiments that match the control panel settings. As with standard topic modules, self-test performance results are reported to users in graphic form (Result dialog box) and are also appended to the cumulative text record of performance in the Results window.

8. DATA PREPARATION

The previous chapters have dealt primarily with the user interface to the CASHE CD-ROM: how the information is presented to the user and what the user can do with it. This chapter and the next one discuss, in very general terms, the data structures and software that support this functionality.

Many decisions must be made in converting the content of a printed document into electronic data, including how to create the digital files, what file formats to use for data storage, how to code data for on-screen display, and how to handle control information.

CASHE has several characteristics that set it somewhat apart from other database products. First, users are expected to employ the product as a tool to guide specific design decisions in real-world human-machine environments. Second, CASHE version 1.0 serves an R&D role in testing an approach for producing interactive multidocument databases of ergonomics data. Finally, CASHE is part of a larger program aimed at developing ways of integrating the use of human perceptual and performance information into an automated system-design environment.

Because of these considerations, we kept in mind the following general goals in selecting methods and procedures for converting the CASHE documents:

- **High accuracy**—the electronic data must accurately reproduce the content of the printed documents.
- **High-quality output**—the data should support both a clear on-screen display and a crisp printout.
- **Structural consistency**—the organization of the data must support the desired interface functionality.

- **Ease of revision and data reusability**—it should be easy to update the documents in the database and to reuse the data in future CASHE products.
- **Portability to other platforms**—conversion of the data to other platforms (such as work stations or PC computers) should be easy to accomplish.
- **Extensibility**—the data structure should be broadly applicable to make it easy to add new documents and new enhancements.

Most of the data files on the CASHE CD-ROM hold the basic content of the database documents—the words and pictures that represent the information to be communicated. As described in earlier chapters, the information in the CASHE documents is organized as a set of short entries focused on specific topics. The information in each entry is subdivided by data class into a text component, one or more figure components, and one or more table components. Each text component, each figure component, and each table component comprises an individual data file and is stored separately on the CD-ROM. Because each class of data presented somewhat different problems during the data preparation process, we will discuss the creation of text data files, figure data files, and table data files separately.

In addition to the basic entry component files, there are separate sets of data files containing the *EDC* Glossary, *EDC* Design Checklist (a set of design-related questions that help users locate relevant *EDC* entries), the access outlines (tables of contents, back-of-the-book indexes, Integrated Outline), and the audio and visual demonstrations. We also briefly discuss how these data files were created.

Finally, the CD-ROM also contains component indexing files that support data retrieval and hyperlinking, as well as text-indexing files that support full-text search. These files are not directly accessible to users but are employed by the engine to organize searches and retrieval of entry components.

8.1 Text

8.1.1 Data Structure and Format

A key strategy of information management in CASHE that serves the goals set forth above is the separation of form from content. This strategy was implemented in preparing the text data files by the use of the Standard Generalized Markup Language (SGML) to code the files for display and printing.

SGML is a method for describing the structure of a document using a standard notation. SGML segregates the information specifying the form of a document (its layout and presentation characteristics) from the information comprising the content and structure of the document. In SGML, a file containing the basic content of a document is coded with descriptive tags that mark the beginnings and ends of logical document elements, such as headings, subsections, or captions. Tags are also used to code nonstandard symbols that may require special rendering (such as Greek letters or mathematical symbols). In addition, tags can point to external entities, such as graphic or sound files.

The tagged file is accompanied by a document type definition (DTD). A DTD is a list of allowable objects and the tags used to identify them. The DTD also specifies the relationships between information elements (e.g., subsections are contained within sections). The CASHE DTD includes the tags required to identify standard structural elements in entry components (entry title, subheadings, reference lists, captions, etc.). It also incorporates several ISO libraries for standard elements and symbols (Greek, Latin, publishing, technical, mathematical, numeric, special graphics, and diacritical marks).

SGML tagged files are pure ASCII files with no formatting information. Instead, processing instructions for each tagged object type are stored separately in a style sheet. Style sheets specify how the information identified by a given tag is to be rendered at run time. For example, the style sheet may define the type style, size, and font in which to display a string of text

tagged as "chapter title" and indicate whether it is to be centered or positioned flush left. Formatting instructions can be different for on-screen display and for print-out, or for different display contexts.

Because it separates content (the tagged ASCII file) from form (the style sheet dictating presentation format), the use of SGML provides a powerful and flexible system for computerized text processing. The style sheet can easily be revised to change the look of a document with no need to edit the tagged files. The tagging system is adaptable to all types of text material. Finally, because the basic data files are pure ASCII, they can easily be ported to new software environments and platforms, where the tags can readily be interpreted by appropriate software. Use of this system for the CASHE database text thus serves the CASHE goals of portability, extensibility, and data reusability.

8.1.2 Data Preparation

From the time that development first began on the *EDC* in the 1980s, the intent was ultimately to make the document available in electronic, as well as printed, form. The printed version of the *EDC* was typeset on computer, and an electronic copy was maintained for an earlier CD-ROM version of the document. However, stripping the proprietary codes from these files to obtain a clean electronic version for CASHE proved much more difficult than had been envisioned. Ultimately, it turned out to be more cost-effective to re-key the *EDC* text for the CASHE CD-ROM than to process the existing electronic files. The other database document in version 1.0, *MIL-STD-1472D*, was available as an electronic file in Microsoft Word format and was converted to ASCII using Word.

All text was marked up in accordance with the CASHE DTD. Since the *MIL-STD-1472D* body text was already available in usable electronic form, this text was tagged in house as a test of the CASHE DTD. At the same time, the text was segmented into the "entry" divisions that were imposed on *MIL-STD-1472D* to make it more usable in the CASHE context. The body of the *EDC*—which, at approximately 2510 pages, constitutes the bulk of the CASHE

text—was keyed in and tagged by an outside contractor, who was provided with the CASHE DTD, instructions for applying the tags, and a custom spelling dictionary.

Each database text file consists of the actual discursive text of the given *EDC* or *MIL-STD-1472D* entry from the printed version, tagged according to the CASHE DTD. The text file also contains additional structural and contextual information for that text component, such as window banner, identification labels, short entry title, and topic section/subsection numbers and titles for *EDC* entries. These tagged elements, which were added during the text rekeying, enable the text component to be displayed accurately and provide the correct content for the viewer window identification region and for the component listing in the History and Window menus as well as in displays such as query results lists or annotation lists.

All of the hypermedia links in the text were specified in the SGML markup. These included hyperlinks to figures and tables, reference links to source citations, and cross-reference links to other entry components, as well as the hyperlinks to next and previous entries used by the entry palette. The tagging identified the source location of the hyperlink, the type of hyperlink, text for the link marker, and the destination component.

The *EDC* contains many special characters, such as subscripts, superscripts, and Greek letters, as well as a number of mathematical equations and expressions, some of them fairly complex. Where possible, special characters and mathematical material were handled as tagged text, with special mark-up codes to render the characters needed.

In some cases, however, the mathematical expressions were too complicated for a mark-up approach to be feasible. In these cases, the equations were handled as graphics. Each equation was produced separately using Expressionist, a mathematics software package by Prescience. This software allowed each equation to be created by hand and saved as a graphics file in PICT format (the standard Macintosh format for graphics). During later

production steps, these PICT files were processed so they appeared embedded in the text at run time. About 120 formulas and equations were handled this way.

Over the years of use of the printed *EDC*, an errata file of typographical errors and other minor revisions had been compiled. These corrections were all entered into the electronic version of the entries. A few additional text changes were also made to accommodate differences in the way figures were handled in the electronic version (e.g., the re-labeling of figures that were split into more than one graphic for electronic display.)

The electronic files for *MIL-STD-1472D* were similarly updated with changes and corrections to bring them into conformity with Notice 3, the most recent release of this document.

8.1.3 Quality Control

Because the CASHE database will be used as an important resource in the design of many military and commercial systems, assuring the correctness of the information was critical. During the production of text files, a sample of text components was proofread carefully for keying errors. Author/Editor (a text-editing software package by SoftQuad that can interpret SGML tagging) was used to display the text on screen for proofing. The typographical accuracy of the re-keyed text for the *EDC* was extremely high.

The SGML tagging was also checked for accuracy and completeness. Author/Editor software was used to verify that tagging adhered to the CASHE DTD and that correct tag syntax was followed. In addition, custom software called StringSeeker was written in house to examine and verify tag content. Each tag was individually analyzed and processed. Tagged strings were checked for internal consistency (for example, within a given text component, tag strings that incorporate the entry number, such as the window banner, entry title, and figure and table call-outs, were all checked to make sure they carried the correct number). The software was also used to tally text

subsections, figure and table references, etc. This information was then cross-checked with the printed entry to make sure all necessary elements were included in the electronic file (for example, to check that no text subsections had been inadvertently skipped). Destination files called out in hyperlink tags were compared with source file lists to ensure that destination designations were accurate and that all destination files existed.

8.1.4 Post-Processing

At the end of the text input process, there existed a marked-up ASCII file for each of the approximately 1150 text components of the database documents. These files were post-processed to integrate the mathematical formulas and other embedded art that had been produced separately into a single, structured file that could be interpreted by the TextViewer to generate a complete and accurate screen display. This post-processing was performed by custom "text blender" software. Input to the text blender was the set of SGML-tagged ASCII files containing the entry text itself, and the PICT files containing mathematical formulas and other graphics to be embedded within the text.

The blender created an output file that basically echoed back the input SGML file. However, if the file contained a tag that designated a formula or embedded art, then the blender located the PICT file specified in the tag, inserted that image as a numbered PICT resource in the resource fork of the text file, and replaced the original tag with a new tag containing the PICT resource ID number.[1] The output text files remain pure ASCII, although their associated PICT data is carried along in the file resources. The final blended text file is stored on the CD-ROM. It is read and interpreted by the TextViewer

[1] All Macintosh files have two segments, a data fork and a resource fork (either of which may be empty). The data fork contains a sequence of bytes representing text, graphics, or other material written into the file by an application and generally accessible in some form to the user. The resource fork contains objects used by an application, such as menus, fonts, icons, or executable code.

to display the entry text component on screen as well as to access any equations or other embedded graphics and display them appropriately in the correct location.

8.2 Figures

Both reference sources on the CASHE CD-ROM rely heavily on figures and illustrations to communicate important information. Because of the central role of graphics in these sources, special care was given to converting the scientific and technical figures contained in them into electronic form. Poor-quality electronic reproduction that degraded or obscured important details of the figures could have made it difficult or impossible for the figures to serve their intended purpose.

Below, we explain how the figures from the CASHE documents were prepared for electronic display. First, we describe in detail several figure display methods that were developed specifically for CASHE to assure that each figure would be readable and interpretable on screen. Then we outline the production process that was used to create the computer data files for the figures.

8.2.1 Figure Design[1]

The two documents included on CASHE version 1.0 contain between them roughly 1500 figures. In the *EDC*, most figures are line graphs plotting human performance data. In *MIL-STD 1472D*, the majority of the graphics are pictorial illustrations depicting government-mandated spatial dimensions for controls, equipment, and work spaces or official anthropometric measurements.

---

[1] This section has been adapted from J. E. Lincoln and D. L. Monk, "Displaying Scientific Graphics on Computer," 1997, *IEEE Transactions on Professional Communication, 40*, pp. 78-91.

Both of the original documents were issued in an 8.5" by 11" format, so that many of the printed figures were quite large. Because we wanted the CASHE database to be accessible to users with standard 13" monitors, the electronic figures had to fit within the content region of the FigureViewer window at its default size—a fairly small display area of about 7" by 5.25".

Approximately one-quarter of the figures in the database would not fit within this designated viewing area in their printed format because they were simply too large, contained too many panels, or were too detailed or densely labeled. For instance, one figure in the printed *EDC* contained 13 separate panels—12 data graphs plotting vibration comfort limit contours for different axes of vibration and body-vehicle contact points, as well as a drawing of a human figure clarifying vibration axes and measurement locations. Another printed *EDC* graphic, an anatomical drawing of the human eyeball, had 55 text labels and over 25 sets of lines and arrows delineating various anatomical and optical features.

Simply shrinking these figures to fit within the available screen area was not a realistic solution. The low resolution of conventional CRT displays (typically about 72 dots per inch) makes it impossible to reproduce very fine details on screen. In fact, some of the smaller figures had to be enlarged to display legibly because of the density of information.

Chopping oversized graphics into viewport-sized pieces without regard to content was no solution either, since it could also destroy their usability and make it impossible for the user to perceive the relationships the figure was designed to convey. For example, dividing the 13-panel vibration figure into 13 separate graphics would make it much harder for users to compare different aspects of the data, such as whether horizontal or vertical vibration is tolerated better, or whether vibration of the seat pan is more uncomfortable than vibration of the footrest.

Instead of arbitrarily splitting up complex or oversized figures or simply "shoe-horning" them into the available display area, we looked for ways

to reconfigure these graphics to preserve their original functionality. After reviewing and analyzing all of the oversized and complex graphics in the database documents, we were able to define a small set of hypertext-based methods for displaying these problem figures on screen to improve their readability and usability as electronic figures. These methods use different ways of parsing information into complementary graphical units that can be displayed separately without loss of context and meaning. In fact, in some cases, reconfiguring the original graphic actually enhanced its usability.

Each of these general display methods is described in detail below. To make it easier to understand the methods, however, we will first review the general figure structure developed to support them.

8.2.1.1 General Structure of Figures: Each figure component in the database is organized as a set of one or more base panels and a caption element (see Fig. 29). A base panel is a data graph, drawing, photograph, or some other portion of the graphic that is displayed as a single unit in the

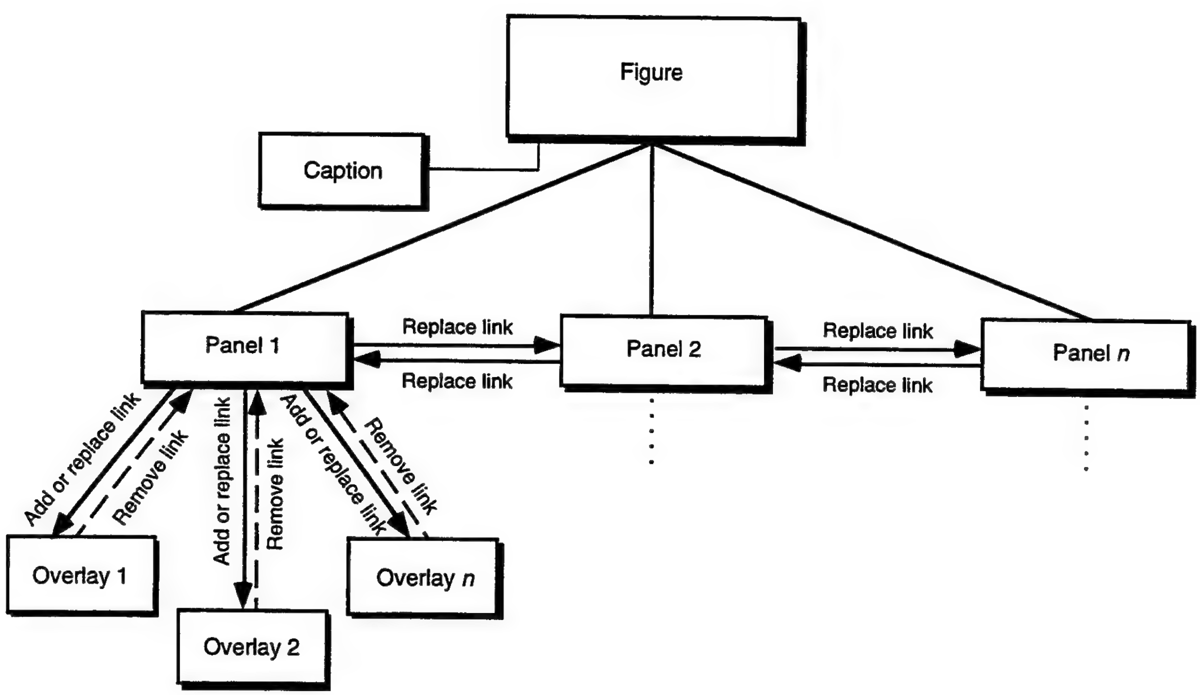


Figure 29. Structure of figure components in the database

FigureViewer content region on screen. A figure may contain more than one base panel; however, the panels are exclusive—that is, only one panel may be displayed at a time.

Each base panel of the figure may have one or more overlays. An overlay can contain text, data points, curves, or other graphic elements. Overlays have transparent backgrounds and are displayed by stacking them over the base panel like acetates. Overlays are sometimes exclusive (only one overlay can be displayed at a time) and sometimes inclusive (as many overlays as desired can be stacked simultaneously).

Base panels and overlays are joined together by hypertext links that allow the user to configure the graphic displays as desired. Four types of hyperlinks are supported: add, replace, remove, and reflexive links. "Add" and "remove" links operate on overlays. An "add" link displays the destination overlay on top of the base panel (and any other overlays already present). A "remove" link removes the source overlay but leaves the rest of the display (including other overlays) unchanged. "Replace" links operate on either base panels or overlays. A "replace" link removes the source panel (or overlay) currently in view and displays the destination panel (or overlay) in its place. "Reflexive" links operate on panels and link the panel to itself. Although reflexive links are required for technical reasons, the user is unaware of them, because activating them does not change the display.

Link markers are employed to alert users to the presence of a hyperlink and to indicate the location of the "hot spot" that must be clicked to activate the hyperlink. Link markers can be buttons, icons, or an embedded region of the artwork. They can be located on any panel or overlay, though of course they will be visible only when the panel or overlay is being displayed and when they are not obscured by a higher overlay.

Because only one base panel of a figure can be viewed at a time, one panel is designated as the "default" panel and is displayed whenever the figure is opened. If a figure panel has overlays, one or more overlays may also be

designated as defaults; these overlays will then be displayed along with the base panel when the figure is first called up.

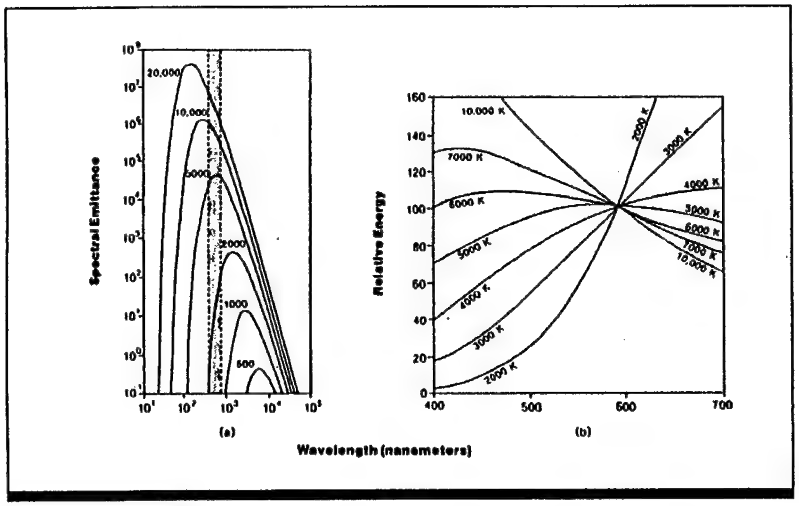
8.2.1.2 Figure Display Methods: The simple building blocks described above—base panels, overlays, and the hyperlinks among them—can be organized in several different ways to allow interactive display of complex scientific figures. The following sections describe these methods.

Split Panels. The easiest and most obvious approach for dealing with a multiple-panel figure is to draw each panel as a separate graphic and display each as an individual figure. This simple method worked well for printed figures with two or more panels that were fairly independent of one another. Figure 30 shows a graphic that was handled in this way. Although the two panels of the original figure are related (both deal with color temperature), each remains completely understandable when it is presented alone.

Each new panel became a separate figure component with its own figure number and caption. To maintain consistency with the printed version, figures that were split were numbered Figure 1a, Figure 1b, and so on, instead of being renumbered as Figure 1, Figure 2, etc.

Merged Panels. Sometimes, several printed figure panels with identical axes could be merged into a single panel by replotting the data curves from all the original panels onto a single set of axes. This method was suitable only for very simple figures that had no more than two or three panels and only one or two curves per panel. Figure 31 shows an example of a two-panel figure handled using this method. Because each original panel had only a single curve and the curves remain well separated spatially when they are superimposed, the resulting composite figure is still quite readable.

Data Overlays. Most multiple-panel figures consisting of data graphs with identical axes were too complex to allow the graphs to be merged together as described above. However, many of these could be rendered as a base panel



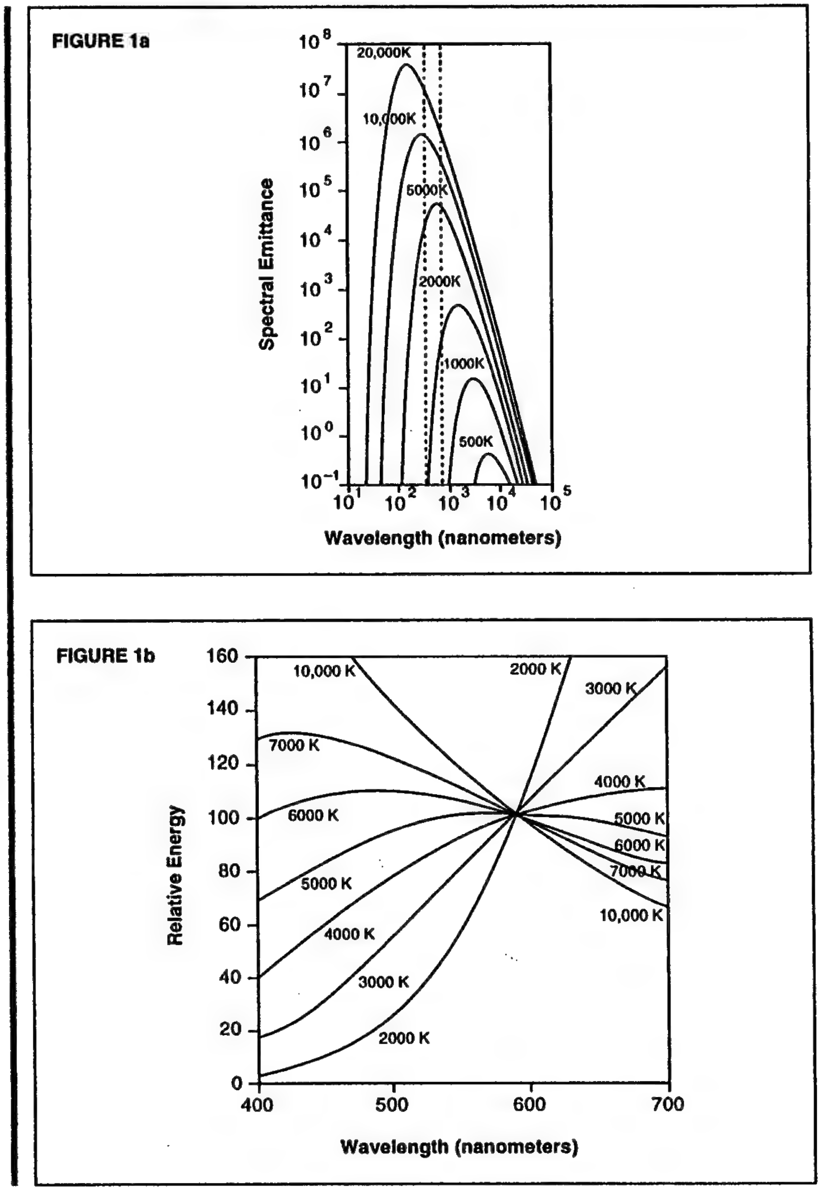


Figure 30. Application of split-panels method. Left: original printed figure. Right: electronic version, in which each panel of the original graphic is presented as a separate figure.

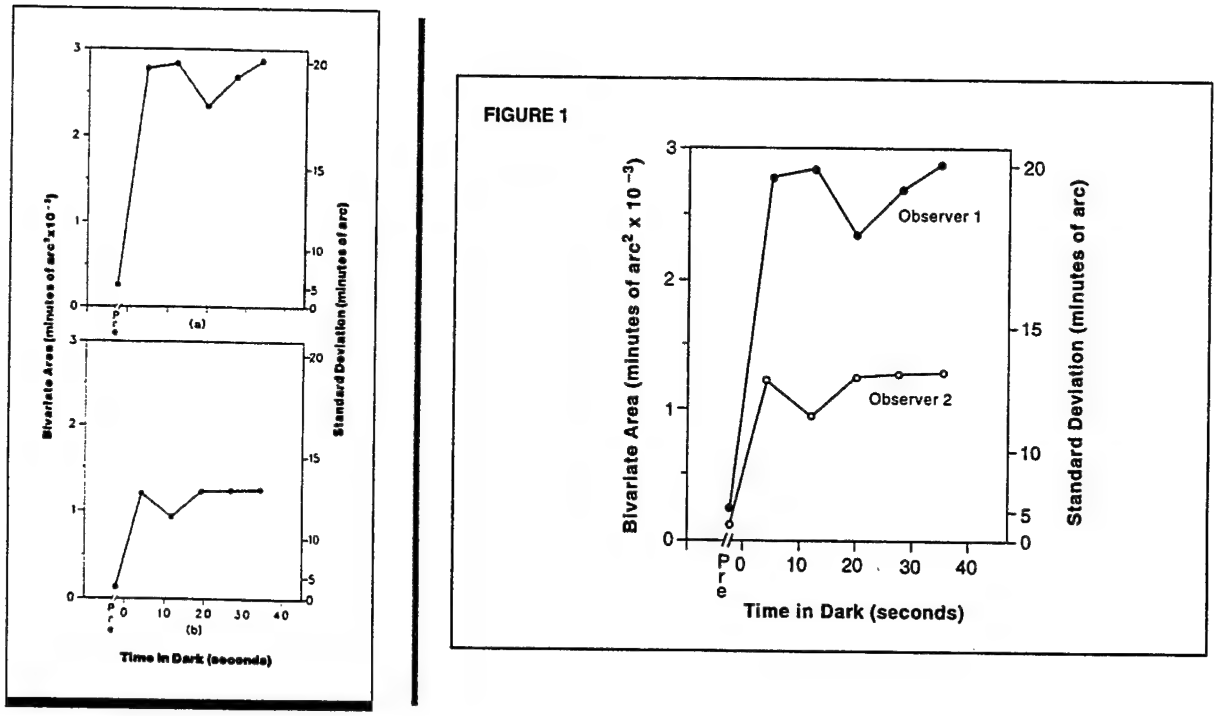


Figure 31. Application of merged panels method. Left: printed figure with two panels showing data for two different observers. Right: for electronic display, printed panels are merged into a single panel by redrawing both data curves onto a single set of axes.

with several data overlays. For example, a printed figure with three panels plotting identical eye focus data for three different subjects was portrayed on screen as a base panel containing an axis box and axis labels, accompanied by a set of three transparent overlays, each showing the data curve for one subject (see Fig. 32). The user can view each data overlay separately or stack overlays as desired to compare data for different subjects. Functionality is actually improved over the paper version, since the user can see the different sets of data superimposed instead of comparing them across separate panels, yet can still view each data curve separately if desired.

The base panel contains only the axis box and axis labels. The base panel is linked to each overlay by an "add" link that opens the corresponding overlay on top of the base panel. Each overlay contains a "remove" link that closes the overlay when activated. Link markers are arranged at the top of the figure as a “menu” of available options. Because they operate as toggles that

turn overlays on or off, the link markers are portrayed as check boxes, toggle-type controls that are already familiar to most users. Clicking an unfilled check box ("add" link) opens the corresponding overlay, while clicking a filled check box ("remove" link) removes that (and only that) overlay when the user no longer wants to display it. The check box labels also serve as a legend for the stacked data curves. Users can simultaneously stack as many of the available overlays as desired (or remove all of them). All overlays are displayed along with the base panel whenever the figure is first accessed (i.e., all overlays are defaults).

Explanatory Overlays. Some figures were difficult to present on screen because they were very densely labeled with long definitions or explanatory material. These figures were also drawn as a single base panel with overlays. Here, the opaque base panel contained the central graphic, minus the long descriptive material or definitions. This supplementary material was put into one or more transparent overlays that appeared on top of

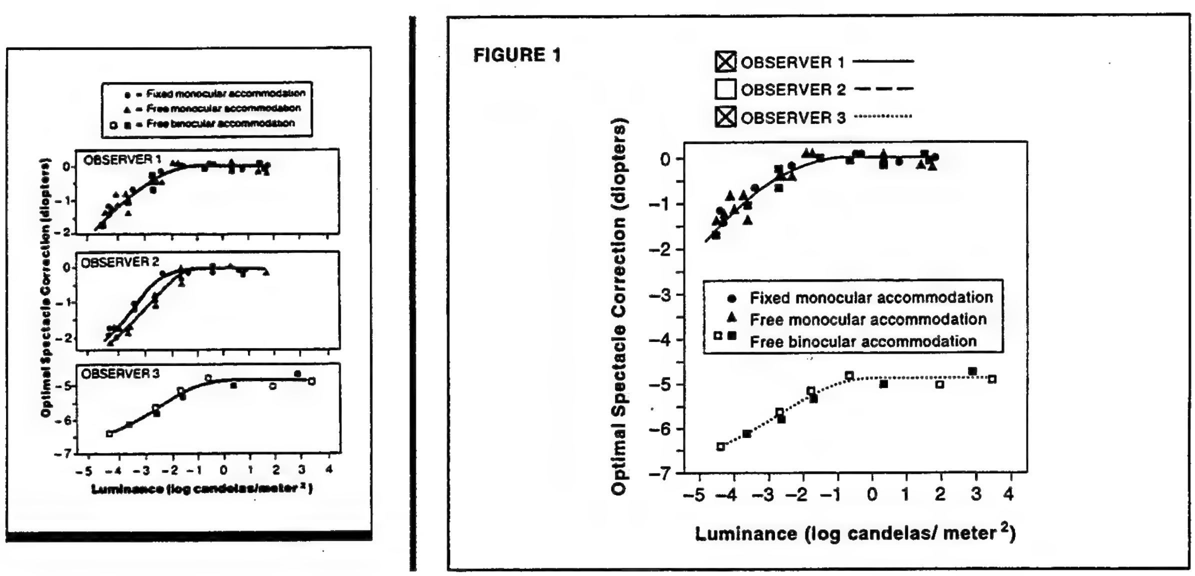


Figure 32. Application of data overlay method. Left: printed three-panel figure. Right: electronic figure, drawn as a base panel with three overlays, each containing the data curve from one original panel. The first and third overlays are currently displayed, as indicated by the filled "check boxes" at the top of the figure.

the base panel like pop-up boxes when the user clicks a specified region or label in the figure.

The base panel is displayed when the figure is opened. There are no default overlays (i.e., no overlay is opened when the base panel is first accessed). To display an overlay, the user clicks an embedded link marker in the base panel, which is generally a label or specific region within the artwork. The overlay is closed by clicking in the overlay to activate a "remove" link. An on-screen prompt guides the user in how to operate the overlays. Only one overlay can be viewed at a time. Clicking a new link marker in the base panel when an overlay is already in view removes the current overlay and displays the new one (i.e., the links from base panel to overlay are "replace" links).

Figure 33 shows a graphic that was handled using the explanatory overlay approach. The original figure was a fairly complex Venn diagram with a definition of each region embedded in the figure. In the on-line version, only the diagram itself and the key for its three major components are drawn in the base panel. The user can click any region of the diagram and display a text box that defines that region and explains its significance. The area the user clicked is rendered in reverse video to signal the region to which the text box applies. The overlay can be closed by clicking this highlighted region (which is actually part of the overlay) or by clicking anywhere in the text box. A prompt at the bottom of the figure alerts users to the availability of the overlays and explains how to access them.

Linked Panels. Figures containing several panels of distinct but closely related graphical information were presented as a connected set with embedded hyperlinks to allow users to move easily from panel to panel. For example, a linked set might include two panels plotting the same data in two different ways, or a set of panels plotting the same type of data for several different target sizes. Instead of simply breaking these multiple panels into separate figures, they are linked to emphasize their interrelatedness and to alert users to the presence of different views of the information. Linking also

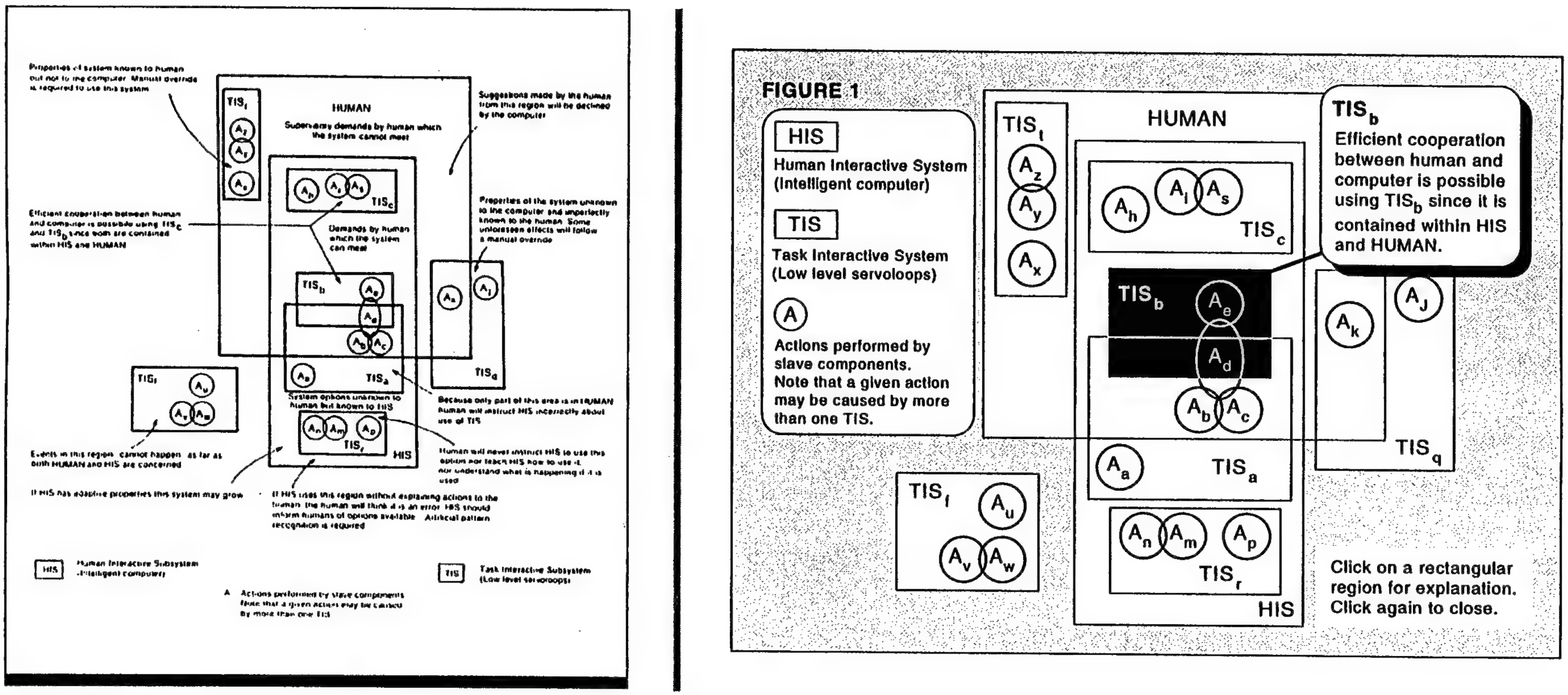


Figure 33. Application of explanatory overlay method. Left: printed figure. Right: electronic version, showing the display after the user has clicked on the rectangle labeled TIS$_b$ (highlighted). The small drop-shadow box is an explanatory overlay that defines the meaning of the highlighted region. The overlay can be removed by clicking anywhere in the drop-shadow box or the highlighted region.

speeds users' access to different panels by eliminating the need to return to the figure menu to switch among panels.

The user selects a panel for viewing by means of embedded controls arranged as a "menu" of panels at the top of each figure panel. This "menu" resembles a standard radio button control list. Each panel has one filled radio button (indicating the panel currently in view) and one or more unfilled radio buttons (identifying other available panels). Clicking an unfilled radio button activates a "replace" link that removes the current panel and displays the selected panel. (Clicking a filled button invokes a "reflexive" link and does not change the display.) Only one of the panels may be displayed at a time, but the user may move freely among them by clicking the appropriate radio buttons.

The entire set of linked panels is treated as a unit and is given a single figure listing in the figure menu for the entry. One panel of the set is designated as the default and is the panel displayed when the figure is opened. The other panels in the linked set are accessible only through the radio-button "menu" at the top of each panel.

Figure 34 shows a graphic that was rendered using the linked-panels method. The two panels of the original figure plotted the same set of skin sensitivity data in two different ways—one panel on linear coordinates and the other panel on log-log coordinates.

One large, extremely dense figure in the *EDC* was actually made much more usable by adapting it to this display method. The original figure showed a human eyeball densely labeled with various anatomical and optical structures and dimensions. For the electronic version, three individual linked panels were created. The same basic drawing of the eyeball was repeated in each one. However, the labeling was divided into three cohesive groupings focusing on different aspects of the information and was distributed among the three panels. One panel defined the basic structures of the eye, another listed the anatomical dimensions of various structures, and the third portrayed the optical constants of the eye. Because the amount of information in each panel

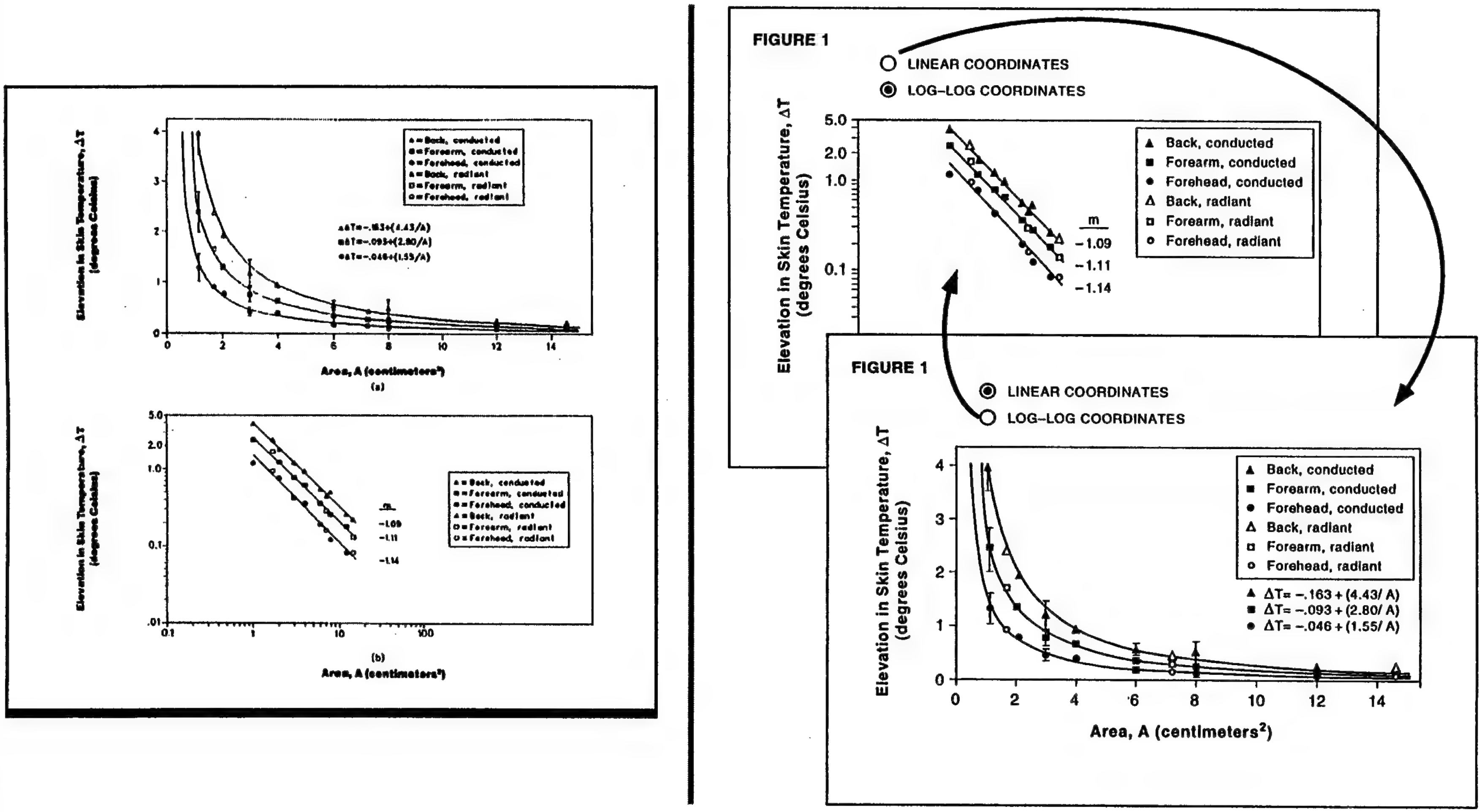


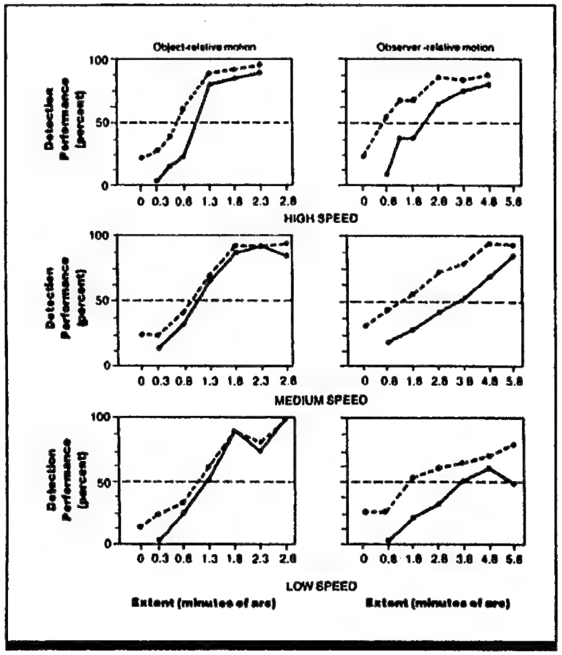
Figure 34. Application of linked panels method. Left: printed two-panel figure. Right: electronic display, in which figure is drawn as two separate linked panels, only one of which is displayed at a time. The filled radio button indicates the panel currently in view. The user removes the current panel and displays the alternate panel by clicking the unfilled radio button.

was reduced, the labeling was easier to read and the user could focus more readily on a specific kind of information about the eye.

Preview-and-Zoom Panel Sets. Some multiple-panel figures were too complex or had too many panels to be suitable for a data-overlay or linked-panels treatment. To display these figures, each individual panel of the multiple-panel figure was drawn up as a separate on-line graphics panel. In addition, a special "preview" or index panel was created to serve as a selection device. Figure 35 provides an example of such a figure. The original graphic was a complex six-panel figure showing subjects' motion detection performance for two types of motion and three motion speeds. In the restructured version used on-line, there is a preview panel containing thumbnail sketches of all the available data panels. The preview provides an overview of the available data panels and allows users to make general comparisons among the data in the different panels. The thumbnail panels in the preview are also link markers. When the user clicks one of these reduced panels, a zoomed, full-size version of the corresponding data panel is displayed. Navigation buttons in the lower right corner of each data panel invoke hyperlinks that allow users to browse through the full-size data panels in order or return to the preview to select a new data panel.

Each set of preview-and-zoom panels is treated as a unit and carries a single figure number. The preview panel is the default panel and is displayed when the figure is opened. Data panels are accessible only via the preview panel. Only one of the figure panels (the preview or a data panel) may be displayed at a time (i.e., all hyperlinks between panels are "replace" links).

Clipped View. This "method" was a fall-back that was used for a very small number of problem figures that were difficult to parse into information units small enough to fit in the allotted display area and could not be adapted to one of the display techniques described above. These figures were drawn full scale and are clipped when they are brought into the viewer window. Users with large monitors can increase the window size so the complete figure is visible. Users with small monitors can scroll to see all portions of the figure or



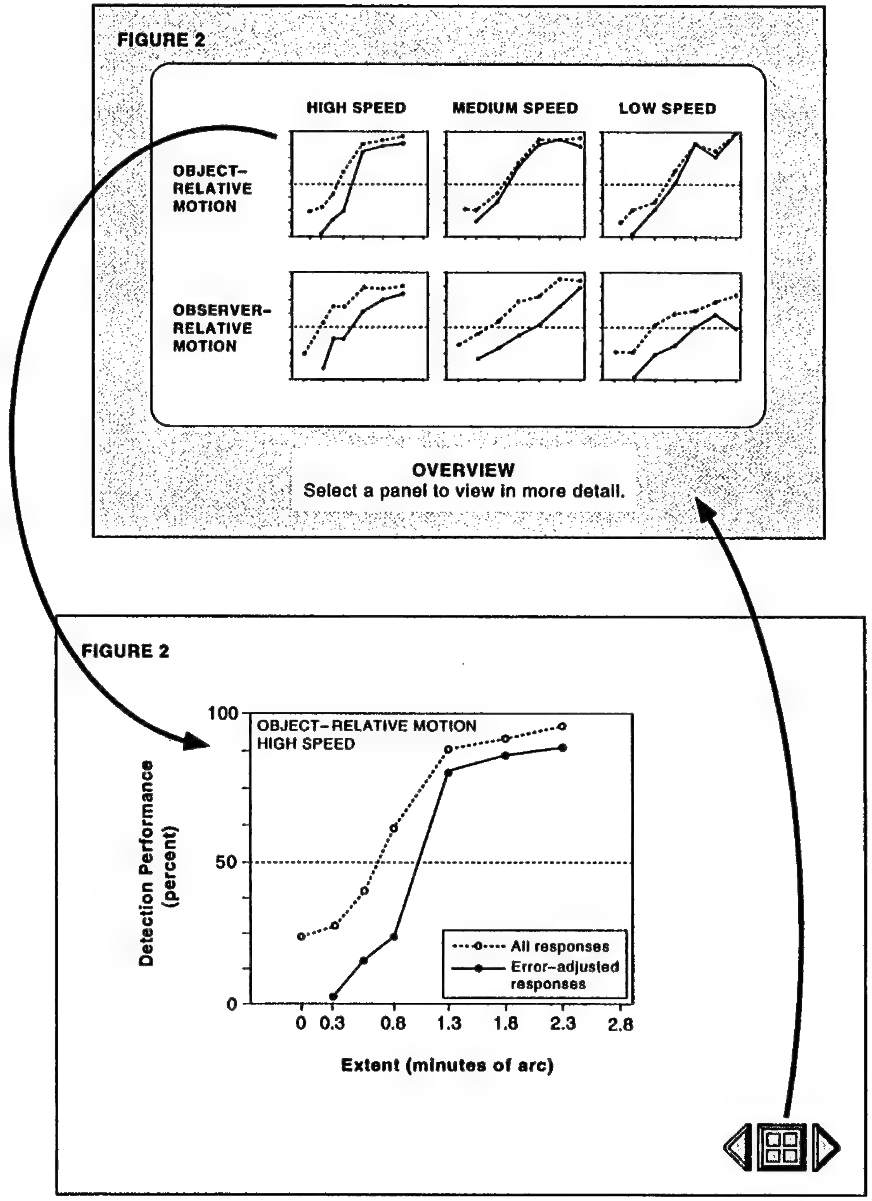
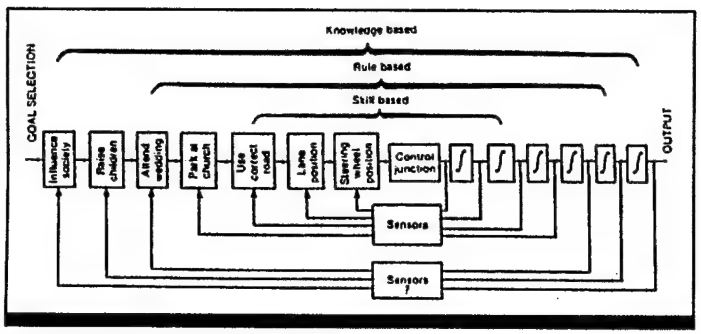


Figure 35. Application of preview-and-zoom method. Above: printed six-panel figure. Right: electronic version. Initial display is a preview panel (upper right) containing "thumbnail" versions of the six data panels. Clicking one of the reduced panel icons displays a full-size version of the corresponding data panel (lower right). Buttons in the lower right corner of the data panel allow users to page to adjacent data panels in the series (arrows) or return to the preview panel (center icon).



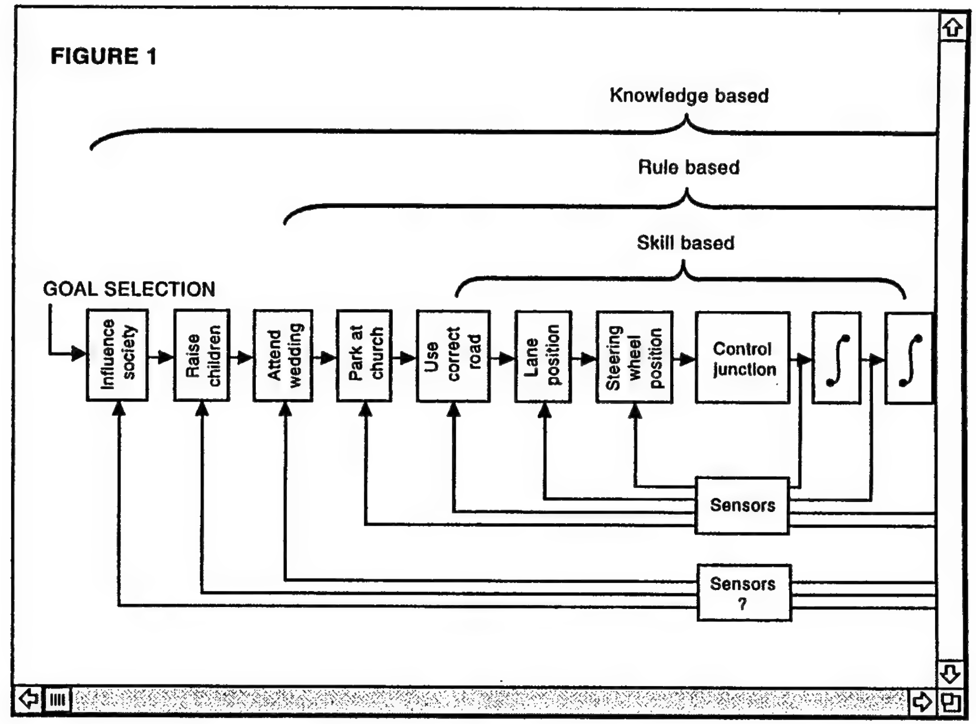


Figure 36. Application of clipped view method. Left: printed figure that was not amenable to any of the special display methods developed. Right: figure is clipped as shown when opened into the FigureViewer at the default window size. Users may scroll to see all parts of the figure, use the "zoom out" function provided in the interface to shrink the figure to fit (some details will be lost), or, on large monitors, enlarge the window to view the entire figure.

can use the "zoom out" function provided by the FigureViewer to reduce the figure so it can be viewed in its entirety, although with some loss of detail. The complete figure can also be printed out. Figure 36 shows an example of a graphic that could not be handled using any of the figure treatment methods described above and is displayed on screen in clipped view.

Combinations of Techniques. Some figures work best on screen when several of the display techniques described above are used together. In *MIL-STD 1472D,* for example, many figures are associated with a table of standard values. Sometimes the table is embedded directly in the figure, while other times it is a separate, independently numbered item. Typically, the pictorial portion of the figure defines and codes various dimensions of control equipment or of the human body, while the associated table provides descriptive or mandated values for each dimension illustrated. These graphics were prepared for electronic display using a combination of the methods described above to increase their usability. For example, one printed *MIL-STD 1472D* figure (see Fig. 37) presents specifications for hand cranks. The pictorial part of the figure defines various crank dimensions and illustrates several crank styles. The tabular part of the figure lists government-mandated minimum, preferred, and maximum values for various hand crank dimensions in both metric and common units and for two different load levels. In CASHE, a combination of the split-panels method, linked-panels method, and explanatory overlay method was used to create a graphic set that not only fits within the standard display area but improves the usability of the material.

First, the original figure was split into two separate figures. The drawings illustrating crank styles, which form a coherent unit and are not necessary for understanding the other elements, were put into one figure. The other figure consisted of the remaining dimensional drawings and table of specifications. These elements were organized as a set of 13 panels joined by "replace" links as in the linked-panels method. Twelve of the panels show identical drawings of basic crank dimensions. Instead of simply illustrating how the dimensions are defined, however, each panel inserts into the

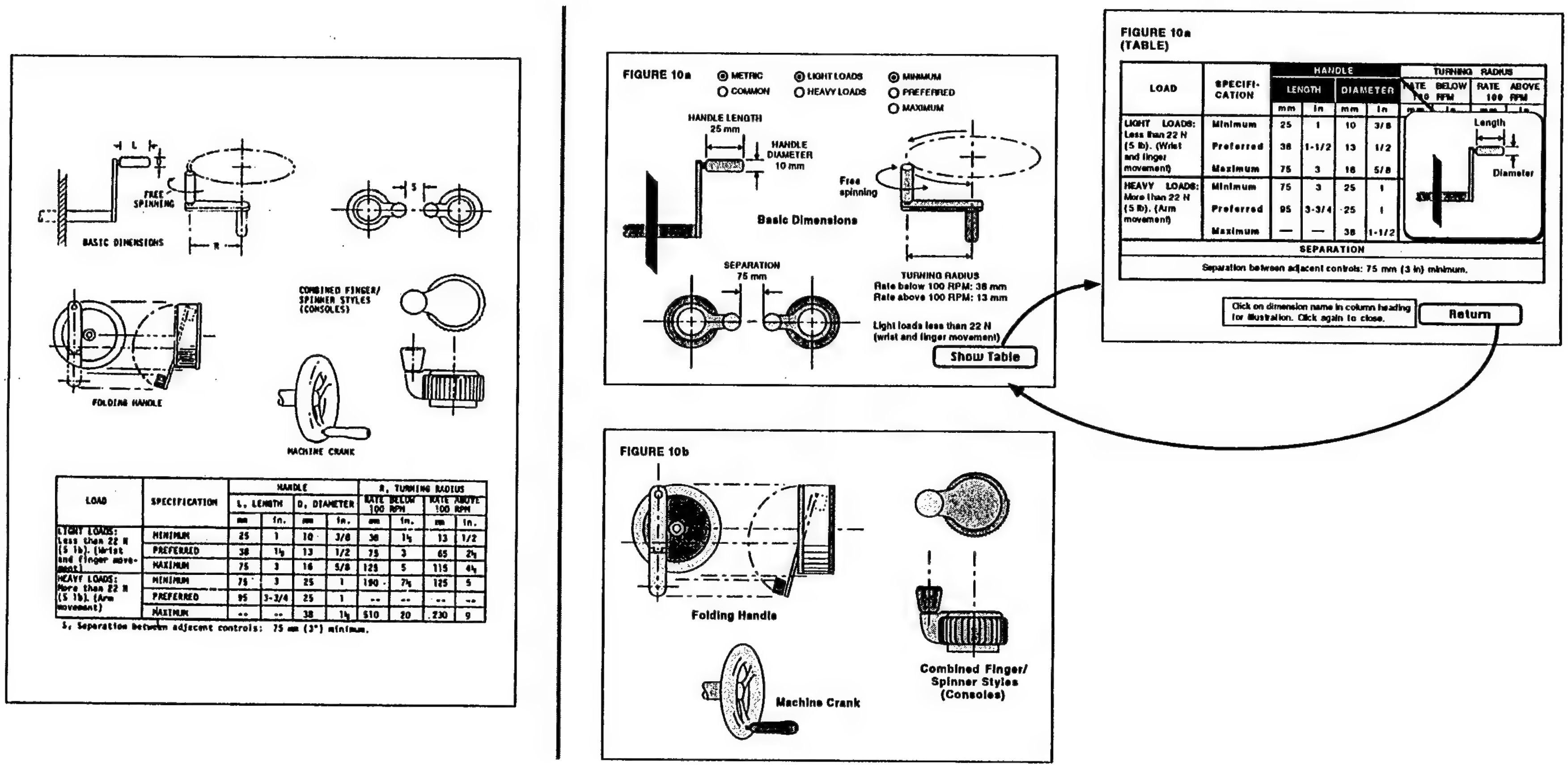


Figure 37. Graphic displayed on line using a combination of display methods. Left: printed figure. Right: electronic display. The graphic is divided into two separate figures (middle panels). One figure (bottom center) illustrates various types of hand-crank controls. The other figure (top two panels) shows control dimensions and their government-mandated values. Top center panel is one of 12 pictorial panels illustrating the official values for various control dimensions. Three sets of "radio buttons" at the top of the panel allow users to specify the exact set of values desired, which determines which of the 12 panels will be displayed. Users can display the full table of values (upper right panel) instead of the pictorial panel by clicking the "Show Table" button. The table panel has explanatory overlays (drop-shadow box with illustration) that define the dimensions listed in the table column headings.

drawings one of the 12 sets of official values for the dimensions from the original data table (see Fig. 37). For example, one panel shows minimum values for light loads in metric units, while another shows preferred values for heavy loads in common (U.S.) units, and so forth. The panels thus serve as filters on the tabled data that allow users to focus on the particular class of values they need. Users can configure the visible figure panel to display the desired data using three sets of radio buttons at the top of each panel.

The thirteenth panel contains the complete table of standard values. The table panel is linked to all 12 pictorial panels and is accessed by means of a "Show Table" button at the bottom of the panel, which executes a "replace" link. The table panel is enhanced with multiple explanatory overlays that help users interpret the tabular information. When the table is displayed, users can click any control dimension listed in the column headings to open a small illustration box showing how that particular dimension is defined (see Fig. 37). A "Return" button at the bottom of the table allows users to return to the pictorial crank-dimension panel from which the table was originally accessed.

In the printed document, users who need to know mandated values for specific control or body dimensions have to shift attention back and forth between the pictorial and the tabular portions of a figure to fully understand what each dimension represents and to locate the specific value corresponding to that dimension. While this is not too burdensome for the figure used in the example, since both the pictorial illustrations and the table of values appear on the same page in the printed version, there are other cases in *MIL-STD 1472D* where the illustrations defining given dimensions and the table containing the mandated values for these dimensions are separated by several pages, forcing users to shuffle back and forth to match definitions with values. By applying the method described here—incorporating the tabular data into the dimensional drawings while preserving the data table as a separate unit with its own pictorial overlays to define individual measurements—the on-line version makes the information easier to use and allows users to obtain all the information they need from either the table panel or the pictorial panels.

Table 3. Methods Used to Display Complex or Oversized Figures

Display Method	Number of Figures
Split panels	128
Merged panels	10
Data overlays	55
Explanatory overlays	11
Linked panels	71
Preview-and-zoom panels	41
Clipped view	4
Combination of methods	36
Total	356

8.2.1.3 Use of the Methods in the Database: Of the roughly 1500 graphics in the CASHE database, approximately 350 would not fit on screen in their original format. To display these "problem" figures, we applied the methods described above. Table 3 shows the distribution of display methods used for these figures. Slightly over a third of these complex or oversized figures could be handled using the simple solution of splitting the original graphic into more than one figure. Still, about 230 problem figures required one of the more complicated display approaches (or some combination of them). Only 4 graphics could not be accommodated by one of the 6 special methods and had to be presented in clipped view.

8.2.2 Data Preparation

8.2.2.1 Image Data Structure and Format: A primary goal in converting the printed figures to electronic form was to assure a high-quality image for screen display and printout. The first major decision in figure data preparation was whether to scan the printed figures or redraw them. Scanning is a fast and relatively inexpensive way to produce digitized images, and is often used when paper documents are converted for computer access.

Scanning is ideal for reproducing photographs and other continuous-tone artwork, as well as illustrations with large areas of fine detail or nonrepetitive patterning. Scanning does not always yield a high-quality reproduction for the type of line art typical of the CASHE documents, however. For example, scans of black and white data graphs may pick up spurious gray tones, and the small type typically used for axis labels and legends may appear fuzzy. Unless careful alignment is maintained during scanning, horizontal and vertical lines and edges may become ragged. Thus, the effort required to clean up the scans may reduce its cost and speed advantages.

Scanned figures also do not re-size well on screen to support zooming functions, may yield low-quality printouts, and may require large amounts of disk storage space. These disadvantages are related to the data format used to manipulate and store the image. Scanning generates a bitmap of the original figure. A bitmap represents an image as a matrix of tiny dots or pixels and records the value of each pixel. The more dots per inch (dpi) used to represent the figure, the greater the resolution and thus the finer the detail that can be reproduced.

Bitmaps produce a high-quality image provided the resolution of the bitmap matches the resolution of the display or output device. Converting an existing image file from one resolution to another degrades image quality, however. One consequence is that bitmapped images displayed on screen can become grainy or blotchy when they are enlarged or reduced in size. Another consequence is that it can be difficult to obtain both a good screen image and a sharp printout of the figure. The reason is the large difference in the resolution of a computer display (70-100 dpi) and the resolution of most printers (300-600 dpi for a standard desktop laser printer). Bitmaps that are optimized for screen display print out poorly at the higher printer resolutions. Bitmaps created at a higher resolution to support good printout may yield an unsatisfactory screen image and require longer to display because of the computations required to step down the resolution.

Another drawback of bitmapped images is the large size of the image files. File size increases with the size of the figure, the level of resolution, and the number of different colors or gray levels supported. A full-screen, high-resolution figure with maximum color depth can require megabytes of storage. Such large image files can rapidly eat up disk space. They also require more computer memory for display and may display more slowly.

Because of these problems with scanned images, we decided to redraw the printed graphics for the CASHE CD-ROM. Although the best screen display can be obtained by drafting the artwork as bitmapped images optimized for the resolution of the display hardware, such bitmaps share most of the disadvantages of scans (poor-quality printout, deterioration with on-screen resizing, and potentially high storage requirements). While good printout quality can be assured by storing two versions of the artwork—one at 72 dpi for screen display and one at 300 or 600 dpi for printout—this substantially increases the disk storage requirements, since a file at the higher resolutions can easily be double or more the size of a file at the lower resolution.

To avoid these problems, we opted to redraw the figures in vector format for electronic use. In vector graphics, a figure is described in software code in terms of basic components or objects such as lines, arcs, rectangles, and ellipses. What is stored for a vector graphic thus is not a representation of the image itself, as with a bitmap, but rather a set of instructions for re-creating the image by drawing the objects that comprise it. The vector graphics approach is especially well-suited for the type of line drawings and data graphs that predominate in the CASHE reference documents. Among its advantages are:

- Screen images remain sharp with vector graphics, even when users resize a figure by zooming in or zooming out.
- Vector graphics print out at high quality at any printer resolution.

- Except for very complex figures, vector graphics files generally are significantly smaller than high-resolution scanned versions of the same artwork and therefore require less disk storage space.
- Alphanumeric text in vector graphics is stored as ASCII strings with attributes, and can be readily located by text-search routines in response to user queries.

Although a vector screen image may be slightly inferior to an optimized bitmapped version, the vector graphics format offers a good compromise for the CASHE product that provides a crisp screen display as well as a high-quality printout and supports the zoom function provided by the FigureViewer.

We selected the PICT file storage format for figure data files. PICT is the standard Macintosh format for graphic data. All major graphics applications for the Macintosh can read and write to this format. PICT files can also be converted easily to other graphics formats, including those common on PC computer systems. Thus, use of this format facilitates future revisions of CASHE as well as expansion of the product to other platforms.

8.2.2.2 Figure Review and Editing: The design of the FigureViewer window allocates an area of approximately 7" × 5.25" for figure display. Before the figure data files could be created, each figure in the printed versions of the *EDC* and *MIL-STD 1472D* had to be reviewed to determine whether or not it would fit satisfactorily within this available screen area.

Two elements were considered in making this judgment: (1) the size of the printed figure; and (2) the complexity (amount of detail) of the figure. Very large figures clearly presented problems, and a decision had to be made as to whether the figure could be reduced sufficiently in size without loss of quality. Figure density was also important, however. The resolution possible with solid ink lines on paper in offset printing is several orders of magnitude greater than the resolution of electronic displays. Thus, details that appear

perfectly clear on printed artwork may disappear or run together on screen because there are not enough pixels to resolve them.

This initial editorial sort identified "problem" figures—complex or oversized figures that would not fit on screen in their printed format and needed special treatment to remain readable and usable on screen. Each of these problem figures was then examined individually to determine which of the special figure display methods described above should be applied. This step was performed by a professional who understood the subject matter to make sure the functionality and tutorial intent of the figures were not compromised.

The reviewer analyzed each problem figure to determine how the user needed to interact with it. What was the purpose of the figure and the primary meaning it was intended to convey? What graphical elements did users need to see on screen at the same time in order to understand and interpret the figure information? What comparisons did they need to be able to make? On the basis of this analysis, a display method was selected that best suited the content and purpose of the figure and preserved its usability.

To assist the artists who would prepare the figures, two sets of instructions were compiled. The first set was general instructions detailing how to prepare figures for each of the seven special display methods. The instructions defined the standard organization of panels and overlays, specified the required link markers and their standard position and presentation style, and outlined any other special requirements to support each display method. (Although link markers and their labels were written into the figure displays by the software at run time, link markers were also prepared as part of the artwork to make hyperlink coding easier.)

In addition to these general instructions, individual instructions were prepared for each problem figure. These item-specific instructions defined for the artist which figure display option was being used for a given figure; how the elements in the printed graphic should be divided up among the requisite panels and overlays; what labels should be used for radio-button,

check-box, and rectangular link markers; and how to position, style, and label any nonstandard link markers or prompts.

8.2.2.3 Copyright Considerations: Most figures in the *EDC* are copyrighted. One important step in preparing the figures for electronic use was obtaining permission from the copyright holder of each figure to include the figure on the CASHE CD-ROM. Although publishers usually allow their copyrighted materials to be reproduced in scholarly works (often for a fee), we encountered difficulties from some publishers in obtaining permission to re-use figures on the CD-ROM. The re-use of copyrighted material on a electronic medium is relatively new, and many permissions departments seemed not yet to have established procedures for dealing with this issue. Publishers were especially concerned about the potential for users to copy and print copyrighted figures from the CD-ROM. We took two steps to ensure that CASHE users are aware of the copyright status of CASHE graphics: (1) the credit line specified by the publisher appears at the end of each figure caption; and (2) the credit is always appended to the figure whenever it is printed or copied from the database document. Even with these safeguards, however, a few publishers restricted reuse of their figures to screen display only. In these cases, printing and export of the individual figure are blocked, so that users can view the copyrighted graphic on screen, but cannot copy it or print it out.

8.2.2.4 Figure Drafting: The electronic version of each figure was prepared by a graphics studio. The studio was provided with a scan of each figure as well as a printed version of the figure. Taking the scans as tracing guides, artists redrew each figure as a vector graphic using Canvas, a commercial drawing program.

Photographs were prepared as high-quality bitmaps. A very few extremely detailed line drawings were also prepared as bitmaps because they contained so many small line segments that rendering the illustrations in vector form actually resulted in a larger graphics file and longer drawing times on screen than did bitmaps of the same art.

Each figure panel and each overlay was prepared as a separate graphics file. Panels and overlays containing link markers were drawn in two layers. The first layer contained the art constituting the graphics panel; the second layer contained only the link markers. This segregation of link information made later production steps easier.

As the figures were redrawn, the figure drafters incorporated the small number of editorial changes and corrections on file for the *EDC* and *MIL-STD-1472D* figures so that the electronic version would be fully up to date.

8.2.2.5 Captions: Because the structure of some figures was changed in the electronic version, the original figure captions had to be reviewed and then corrected if necessary. For example, graphics that had been divided into more than one figure using the split-panels method needed a separate caption for each new figure, and the original caption was split accordingly. References to figures in the running text also had to be checked to make sure they were still accurate. Although the caption and embedded figure references had to be examined individually for each figure that received a special graphics treatment, the total number of changes required was relatively small.

The caption for each figure was prepared separately in house as an individual SGML tagged ASCII file. In addition to the caption itself, each caption file contained structural and contextual information (such as the window banner and entry number/title) required for proper display.

For copyrighted figures, the full source citation and the required permissions credit line were added at the bottom of the caption. A special permissions tag was also added when the copyright holder denied permission to print or copy the figure from within CASHE. Any typographical errors in the captions that had been discovered during use of the printed version were also corrected.

The printed *MIL-STD 1472D* figures had figure titles; the printed *EDC* figures, however, did not. To improve their usability in the electronic

version, figure titles were added for all *EDC* graphics for use in figure access menus, search results lists, etc.

8.2.2.6 Quality Control: Because of the important role played by figures in the reference sources, great care was taken to ensure that the figures were redrawn accurately. The graphics studio proofed the figures before shipment and filled out a quality control check sheet for each figure. The figures were proofed a second time by project staff at Armstrong Laboratory. For figures that required special display treatment, electronic files were compared with the individual instructions for the figure to verify that the correct display method had been applied, that all required base panels and overlays had been created, and that the content of each panel and overlay was correct.

Captions were checked in the same manner as other text files. Author/Editor software was used to proof the captions for typographical errors and test tag syntax and structure. Tag content was then analyzed and verified using custom-written StringSeeker software.

8.2.3 Post-Processing

Additional processing was required to convert the graphics files into a form that could be interpreted by the FigureViewer and contained all the information necessary to display the figures properly on screen. To guide these steps, general specifications for the seven special display methods as well as individual display instructions for each problem figure were prepared. Together, these materials defined the panel/overlay organization and file structure of each graphic, provided a link map and definition of link types for the linked panels and overlays in each figure, and specified default panels and overlays.

When the drafted figures were proofread, two figure databases were created simultaneously. The first database—the hyperlink or "hotspot" database—contained the hyperlink data required to control the presentation of panels and overlays in order to implement the hypertext-based figure display

methods. This database was constructed from the second (hyperlink) layer of the graphics files. This database recorded the following information:

- source of the hyperlink (figure panel or overlay);
- destination of the hyperlink (figure panel or overlay);
- position coordinates of the link marker;
- presentation style of the link marker (check box, radio button, etc.); and
- label text for the link marker.

Once the appropriate hyperlink information for the given figure panel or overlay had been entered into the hyperlink database, the link layer of the graphics file was discarded and the file was saved in PICT format.

The second database—the figure panel definition database—defined the panel and overlay structure of each numbered figure. The elements in this database included:

- the full figure number (which included the entry number);
- the filenames of all the individual PICT files that comprised each numbered figure; these included one file for each panel and one file for each overlay;
- the type of each PICT file (base panel or overlay); and
- the default status of each panel and overlay (i.e., whether it was to be displayed when the figure is first opened).

The last step during figure production was to generate a single, structured file for each numbered figure to be stored on the CD-ROM and interpreted by the engine software at run time to display the figure and execute its internal hyperlinks. A custom-written "figure blender" program assembled this file by merging the data from the four different types of graphics-related files created during figure production: (1) the PICT graphics files (one file for each figure panel and each overlay); (2) the caption files (one caption file per figure), which were SGML files that also included some structural and

contextual information as well as permissions information; (3) the figure hyperlink database, which included all the required hyperlink information for each figure panel; and (4) the figure panel definition database, which identified the specific panels and overlays associated with each numbered figure. The blended file for each figure was constructed as a commented PICT file and contained:

- graphic information for all figure panels and overlays for the figure;
- location of hyperlink hot spots in each panel and/or overlay, directives for drawing the corresponding link markers, and specification of the hyperlink destination;
- text for the figure caption;
- labels and tags (such as the figure number and title, and the number and title of the entry of which it is a component) that are used by the software at run time to fill in the identification region of the window in which the figure is displayed, construct the window title bar, identify the figure in menus, query results lists, and annotations lists, etc.
- permissions flag, which is interpreted by the software at run time to enable or disable the Copy and Print commands in the Edit and File menus.

The blended file was then encrypted to prevent any copyrighted figures from being viewed without the appropriate permissions information. The blended file for each figure is the basic data file for the figure that is stored on the CD-ROM. This file is interpreted by the FigureViewer at run time to display the figure accurately on screen.

8.2.4 Final Check

Once the FigureViewer module of the interface software was available, each blended figure data file was given a final check by opening the figure in the viewer. The panel and overlay structure of the figure was confirmed, the style and placement of link markers was checked, and the destination of each link was verified by traversing the link.

8.2.5 Sidelight on Technological Pitfalls

There is an interesting sidelight on figure production for CASHE that illustrates the pitfalls of being ahead of the technological times. From the first days of the IPID project (CASHE's predecessor), the long-range goal was to provide computerized access to human perceptual and performance data. Thus, the figures for the *Handbook of Perception and Human Performance* (many of which were re-used in the *EDC*) were drafted by computer, with the intent that these electronic files could be used in future products as the project progressed. In the mid-1980s when the *Handbook* was produced, however, computer drafting was still in its infancy. Figure-drafting routines were commonly custom-written software and the resulting electronic files were in proprietary formats not interpretable by other drafting programs. When the *EDC* was compiled two years later, the original electronic figure files for *Handbook* figures proved unusable and the original paper printouts had to be used. Because computer graphics generation seemed to have progressed little by that time and was considerably more expensive than conventional hand-drawn artwork, the decision was reluctantly made to use noncomputerized drafting for the many new figures needed for the *EDC*. By the early 1990s, computer-generated graphics had entered a new era of high-quality, relatively low cost production and standardized, transferable file formats. These advances helped make the development of CASHE feasible—and hopefully will ensure the reusability of the figures.

8.3 Tables

8.3.1 Table Design

As with figures, the design of table displays was approached from the viewpoint of ensuring their usability. The majority of the 450 tables in the CASHE database could be fit within the content region of the TableViewer window. However, a small number of tables (approximately 8%) were too large to fit on screen. Simply clipping these tables into a conventional scrollable

window was rejected as a solution. Although standard scrolling would allow users to bring hidden portions of the table into view, column headings and other context-setting elements could move out of sight, making it difficult for users to interpret visible tabular information.

After considering how users needed to interact with the tables, we decided that the most efficient, easiest to use, and most cost-effective approach was to support a special type of scrolling that preserved context. The column headings and the stub (the rows of the leftmost column) define the context for a given table cell. Therefore, when the user scrolls a table vertically, the column headings do not scroll up out of sight, but remain stationary, while the stub and table body slide under them. Likewise, when the user scrolls horizontally, the stub remains stationary while the rest of the table scrolls against it. This type of "freeze-pane" scrolling assures that the stub row and column header providing identification for a given data cell will always remain in view.

MIL-STD-1472D contains several complex tables that were difficult to display well in the TableViewer on a small screen, even with the special tabular scrolling. These tables typically include several parallel sets of data and either contain embedded art or are closely associated with regular numbered figure components illustrating the dimensions for which data are given. For example, one of the more complex tables lists anthropometric measurements for the head and face and provides 24 different sets of data for each dimension, categorized by gender (male vs. female), personnel group (general forces vs. army pilots vs. air force pilots), percentile (5th vs. 95th), and measurement unit (centimeters vs. inches). This table is intended to be used with a numbered figure that contains drawings of the human head showing how to measure the different dimensions for which data are given in the table.

To display such complex tables in a way that preserves their usability, they are structured using the same types of display techniques employed for figures, as described above in section 8.2.1. The tabular data are divided into several complementary panels, each containing one specific data set (for example, 5th percentile data for male army pilots in metric units, or

95th percentile data for female ground troops in inches). Users select among the multiple panels by means of a set of radio button controls at the top of each table panel, just as with linked figure panels. The table generally links to a numbered figure and/or pop-up graphics that illustrate how to measure the dimensions for which data are provided. For example, the name of a dimension in the table stub (e.g., "elbow rest height") may be a hot spot that, when clicked, pops up a window containing a profile drawing of a seated person that shows how elbow rest height is defined. A standard rectangular button in the title area may also link the user to an independent numbered figure component that defines all the dimensions listed in the table. Figure 38 shows an example of a table handled in this way.

8.3.2 Data Structure and Format

Tables, like text components, were created as SGML-tagged ASCII text. Although we considered treating the tables as graphics and storing them in PICT format, we rejected this approach. Conceptually, tables are much closer to text than to figures. In addition, with PICT tables it would have been more work to implement the special freeze-pane scrolling that typifies the TableViewer presentation and makes the tables easy to use by keeping context-setting elements in view. Also, text searches of the material in the tables would have been less user friendly. Although the search routines could have identified query strings in the PICT file, the search software would not have allowed scrolling to or highlighting of the terms in the tables containing them.

The CASHE database documents contain approximately 500 tables. Most are straightforward, all-text tables. About two-thirds have a very simple structure with only one level of column heading.

The tables were keyed in with the rest of the entry text. Each table was prepared as a separate file and tagged for rough layout. Once the tables were typeset and tagged, each table was viewed individually using Author/Editor software. Column widths and other elements were adjusted manually to provide a clear and pleasing layout. The tagging was then automatically

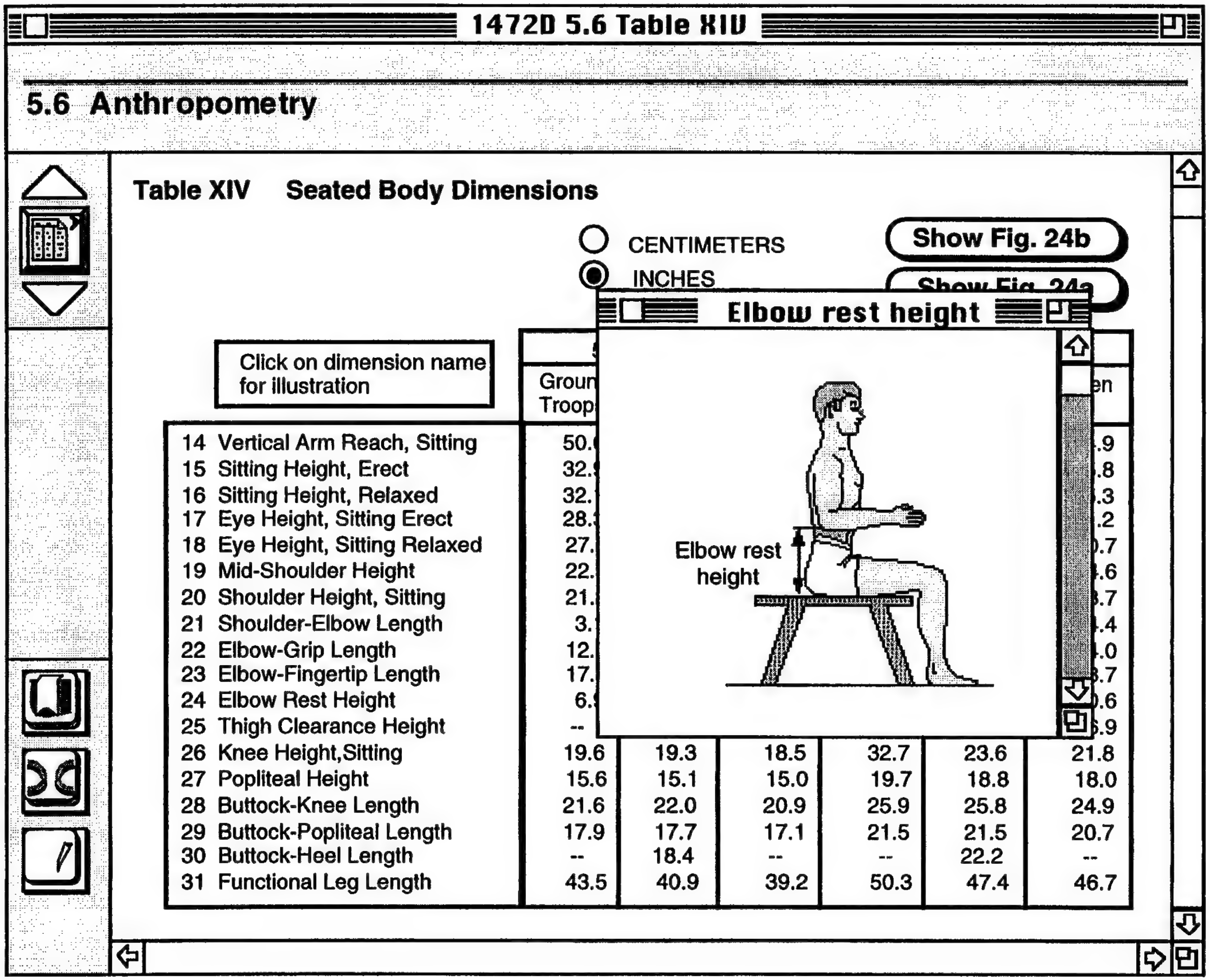


Figure 38. One panel of a linked-panel table showing pop-up graphic that appears when user clicks "24 Elbow Rest Height" line in table stub.

updated by Author/Editor to reflect this final layout. The tagging used to generate the on-screen layout also supported the special freeze-pane scrolling implemented by the TableViewer.

Some *EDC* tables had titles; others did not. As with *EDC* figures, table titles were created for tables when necessary for use in table access menus, search results lists, etc. These titles, along with other structural/contextual information such as window banner content, entry titles, and section/ subsection titles, were added when the tables were keyed in.

Slightly over 10% of the tables contained mathematical formulas or equations, had artwork embedded in the body of the table, or required pop-up graphics accessible from the table. As with the text components, the equations were treated as graphics and produced separately using Expressionist software. The artwork for embedded figures or pop-ups was drafted using a drawing program. Each mathematical formula and graphic was stored in a PICT file. These files were then processed along with the table text files using the custom text blender software. As with the nontable text files, the blender stored each PICT formula and graphic in the resource fork of the appropriate file and inserted coding so that each would be displayed at its proper location in the table.

As with the text components, all necessary information to display and execute hyperlinks in the table components is specified in the SGML markup. This includes the hyperlinks to pop-up graphics and independent figure components as well as hyperlinks to source citations and cross-references to other entries.

Some *EDC* tables contain cross-references to citations in the Key References section of the entry text. The full bibliographic citations for these sources were inserted at the bottoms of the tables so that complete source information will remain with the tables if they are printed or exported. (Clicking the embedded "Ref." cross-reference in the body of the table still links

the user to the corresponding citation in the Key References section of the text component.)

As with figures, some tables reproduced in the CASHE reference documents are copyrighted. Permission had to be obtained from the copyright holder to include these tables on the CD-ROM. Permissions credit lines for these tables were added at the bottom of the table in the footnote area. The appropriate tags to block printing and/or copying were also added to the table file when the copyright holder denied permission to print or export tables from CASHE. Any table corrections in the general *EDC* errata file or most recent *MIL-STD-1472D* notice were also entered. All these additions were made in house after the tables had already been keyed in and tagged.

8.3.3 Quality Control

Because CASHE will be used as a design reference tool, accuracy of the information in the tables is very important. When tagged table files were viewed individually using Author/Editor to adjust layout, the numerical data in each table were also proofread for accuracy. At the same time, permissions credit lines and, when necessary, permissions tags to block copy and printing were added for tables reprinted from copyrighted sources. Tagging was also checked for accuracy and completeness using the same proprietary StringSeeker software utilized for text components.

8.3.4 Post-Processing

The SGML-tagged table files were post-processed using the "text blender" in the same way as the text component files to integrate the PICT formulas and embedded graphics, convert them to PICT resources, and insert the appropriate resource ID numbers into the table files.

Tables with pop-up graphics were processed similarly, except that the artwork for such tables is stored as a PICT resource of a separate "graphics" file, rather than as a resource of the table itself. The table contains a

hyperlink to this external file that, when activated by the TableViewer, results in the opening of a general window to display the pop-up graphic.

8.3.5 Final Check

The blended table files were given a final check once the TableViewer module of CASHE was operational. Each table was displayed on screen using the TableViewer and final adjustments were made to column widths and layout. Most of the tables required some fine tuning to adjust to the CASHE word-wrapping algorithm.

8.4 Other Data Files

The text, tables, and figures from the two database documents in CASHE version 1.0 comprise the bulk of the data files on the CD-ROM. Several other types of data files also had to be created, however. These included the access outlines, Glossary, and Design Checklist (basically text files) and the audio and visual demonstrations.

8.4.1 Text Files

8.4.1.1 Access Outlines: The access outlines include the Integrated Outline (which covers all database documents) as well as the table of contents and back-of-the-book index for each document, and the list of figures and list of tables for *MIL-STD-1472D*.

To create electronic versions of the table of contents and index for the *EDC*, these materials were scanned using optical character recognition (OCR) software. Electronic versions of the Integrated Outline and the *MIL-STD-1472D* materials were already available.

All these files had to be coded with the proper tagging to reflect the hierarchical structure of the material. Entry numbers and titles had to be added at the lowest level of some files, and all hyperlinks had to be coded

appropriately. This work was done in house. The result was a set of tagged ASCII files whose SGML markup enables the OutlineViewer to re-create the appropriate tree structure so the user can expand and collapse the outlines when they are displayed on screen and link to entry components from the lowest level.

The system "bookkeeping" required to keep track of the current state of expansion of every heading in an outline sets a rather small upper limit on the size of file that can be displayed in the OutlineViewer without a severe degradation in performance. For this reason, the longer outlines (the tables of contents and back-of-the book indexes) were segmented into one master outline file and a series of smaller lower-level files. The master file contains only the top level or two of the topic hierarchy, while the lower-level files contain the remainder of the topic tree for individual sections, including the anchor level that links to the entry components.

For example, the table of contents outline for the *EDC* consists of a master outline and twelve subsidiary outline files. When users select the *EDC* table of contents from the Documents menu, the OutlineViewer opens with a listing of the twelve major subject headings of the *EDC* (master outline file). When the user clicks one of the headings, a new outline window is opened containing the full tree structure for that subject heading.

8.4.1.2 Glossary: The printed *EDC* Glossary was also scanned using OCR software. Users can display Glossary definitions through hyperlinks in the entry component text. So that users can also browse the Glossary like a document, a simple two-level browsing outline was created. The top level of the outline is the letters of the alphabet and the secondary level is the list of terms defined in the Glossary. Each term links to its written definition. To improve speed of access and file manipulation, the two-level outline was placed in one file and the Glossary text definitions are divided into a separate series of 11 files. When the user selects a term in the browsing outline, the corresponding Glossary file is opened and scrolled to the correct term. All the files were prepared as ASCII text files with SGML markup added

in house to support appropriate manipulation and display of the material in the OutlineViewer and GeneralViewer.

The Glossary files were enhanced to enable the system to find matches for inexact terms. For example, clicking on the term *astigmatic* in a text component opens the Glossary entry "Astigmatism." To support this feature, first the proprietary program StringSeeker was used to search all text component files and read the content of all items tagged as Glossary terms. Then alternate forms were sorted manually, mapped to the correct (master) Glossary entry, and inserted as entry points into the Glossary files.

8.4.1.3 Design Checklist: The Design Checklist is a hierarchically structured set of questions to assist users in locating *EDC* entries related to specific design problems. Like the Glossary, the Design Checklist consists of a browsing outline (containing the topic hierarchy used to organize the questions) and the text of the questions themselves. Also like the Glossary, the disk files for the Checklist were prepared as a single outline file and a set of 15 text files that contain the text of the questions. Since the Checklist was edited and expanded for the CASHE CD-ROM, the electronic files were produced in house and marked up with the appropriate SGML tagging to support correct display.

8.4.1.4 Text File Quality Control: Once they were created and tagged, all the auxiliary files for the access outlines, Glossary, and Design Checklist were put through the same quality control checks as the other text files. The files were proofread using Author/Editor. Author/Editor and the in-house routine StringSeeker were used to check tags for accuracy and proper syntax and to verify destination components for hyperlink tags.

8.4.2 Audio Demonstrations and Animations

The 71 audio demonstrations and visual animations that link to the database information were another set of data files that had to be prepared for the CD-ROM. Because the demonstrations were dynamic and intended to

illustrate specific effects, brief specifications were written for each item to clarify its operation for the programmers. For the simpler demonstrations, a brief description of the required events was generally sufficient. For others, more detailed storyboards had to be created showing the display for each frame and the overall flow.

The demonstrations were programmed in Macromedia Director. The sounds for the audio demonstrations were pre-recorded or generated digitally using SoundEdit. After programming, each demonstration was carefully examined by a subject-matter expert and other reviewers to make sure the demonstration produced the intended perceptual effects.

The completed demonstrations were converted to QuickTime format. (QuickTime is a Macintosh system extension that provides the capability to use digitized video and audio files.) This conversion allows the files to be played back from the CD-ROM using the Macintosh MoviePlayer (rather than proprietary software).

Playback of animations from CD-ROMs can be jerky because of the time required to read in the new data each time the screen changes. The problem is especially bad for animations with rapid frame rates and can result in the omission of some frames. To assure smooth playback, the QuickTime files for CASHE were post-processed using Apple's ComboWalker to "flatten" the audio/visual animations. Flattening spreads out the loading of data so the data for each frame are available the moment they are needed and the playback is uniform. Post-processed QuickTime files were again reviewed to confirm that all demonstrations operated properly.

8.5 Indexing Files

All of the data files described above are displayed to the user in one form or another during the user's interactions with CASHE. The CD-ROM also contains two other types of data files that are not accessible to the user.

These data files are employed by the system to locate component data on the disk.

8.5.1 Component Index

Because the text, table, and figure portions of *EDC* and *MIL-STD-1472D* entries are stored in separate files, the information pertaining to a given entry is fragmented. To provide the system with a single source of knowledge about the components of an entry, the CD-ROM contains a machine-readable index to the data on the disk. This component index is a database that records which text, table, and figure components belong to a given entry and supplies pointers to the corresponding component files (i.e., indicates their locations on the disk). It also specifies which test benches are accessible from the entry and which test bench topic (if any) should be flagged when the test bench is launched from the entry. In addition to defining the structure of a given entry, the component index records the full entry title, the short entry title (used in menus and search results lists), and the title of each associated figure, table, and test bench; defines the next and previous entries; identifies all the references contained in the entry; and provides various other information about the entry.

The component index also catalogs information about non-entry data files on the CD-ROM, such as the access outlines, Glossary, Design Checklist, and audiovisual demonstrations. It records the "component" structures of all data files and entry points that are possible destinations for hyperlinks and supplies file pathnames, pointers, and offsets so that files can be located on the CD-ROM and hyperlinks can be executed correctly. For example, it identifies which Glossary file contains the definition for a given Glossary term, provides a pointer to the file, and supplies the offset to the definition of the term in the file.

A component database specifying the structure of each entry was generated by the text blender from information in the text component data file at the same time that the text file was processed into its final form. Infor-

mation for the other data items (QuickTime audiovisual demonstrations, access outlines, etc.) as well as information about the test benches and other supplementary information was inserted manually into the component database. This database was then processed into machine-readable form to create the component index.

All data requests generated in CASHE (e.g., a text cross reference, figure or table call-out, or Glossary term) are referred to the component index (via the CASHE Server). The correct filename for the required component (and, if relevant, the offset within the file) is looked up in the component index and passed to the viewer so the component can be opened. Information regarding entry structure is also passed to the entry palette so the palette can be updated. (Interactions with the component index are described further in the next chapter.)

8.5.2 Search Index

When the user invokes the Query command to locate a text string or expression in the CASHE database, the entire text of all database documents must be searched. Such a search would be far too time-consuming to be performed at run time. To speed full-text search, an inverted index of the text of all database documents was created. This index is stored on the CD-ROM (and is copied to the user's hard disk during normal installation).

8.5.2.1 Inverted Index: The inverted index is a list of all the words occurring in the database along with the locations where each word is found. The inverted index was created by a custom indexing program that scanned all the component files.

The CASHE CD-ROM contains a single master inverted index of all text for both documents on CASHE version 1.0. Only the text in the documents is indexed. This includes all the text in text and table components, as well as the text in figure captions. PICT elements (figure graphics and mathematical equations treated as graphics), as well as outlines, contents lists, and the

glossary, are not indexed and thus cannot be searched using Query. The inverted index contains an alphabetic listing of all index terms. Accompanying each term is a list of all the occurrences of the term in the database documents. A component filename and the offset from the top of the file at which the term appears is recorded for each occurrence. Because of the location information that must be provided for each occurrence of a term, inverted indices are generally very large files (they may be twice the length of the original text files).

8.5.2.2 Truncation of Terms: Research by Gerard Salton[1] has found that user queries which search for an exact input recall only about one-third of the pertinent documents, on average. Of the documents that *are* recalled, only half are of interest to the user.

Salton also found that the length of words used in the query matching directly affects the quality of the query results. The strategy he recommended is to automatically reduce the length of the words used in the query. This means limiting not only the length of users' query inputs, but also the length of the words stored in the internal index to be searched.

There are various methods for reducing word length, such as removing vowels, finding the root word, truncating to a given number of characters, etc. Salton found that simple truncation is nearly as effective as the more complex strategies and yielded significant savings of execution time. For example, the query word *visualization* might be truncated to *visual,* which would then match *visualizing, visually,* and *visual* as well. Users were found to frequently overspecify the query words themselves, and occasionally under-specify the combinations of those words (i.e., groupings with AND and OR).

The advantages of shortening words is twofold. First, the recall of the query term improves with smaller word bases. For example, a user interested

---

[1]*Automatic Text Processing* (pp. 245-248), by Gerard Salton, 1989, Reading, Mass: Addison-Wesley Publishing Co.

in querying on the subject of vision is probably interested in finding all of the articles with the words *vision, visual, visualizing, visualization*, etc. If the words are chopped to a small base, the user will probably retrieve a greater number of relevant entries.

Second, truncating words saves storage space. This might allow the word list to fit into RAM instead of being read off disk, thus increasing the speed of the query. For example, if the documents contain 17,000 unique words and the longest word is 20 characters, the word list (excluding location information) would take 340,00 bytes (20 × 17,000). By truncating words to 8 characters, the number of unique words might decrease to 15,000 unique words, since words like *visualization* and *visualizing* would drop to the same base, *visualiz*. Thus, the truncated word list would require only 120,000 bytes of storage (8 × 15,000).

The only disadvantage of truncating words is the loss of information about specific words. For example, a user who really wanted only to look for the word *visualization* would have to wade through returns that also included the word *visualizing, visualized*, etc.

Storage requirements, document vocabulary, and types of user queries must be considered in deciding on word length. Using longer word lengths (8-10 characters) results in fewer query recalls (greater number of pertinent entries missed), higher memory usage, and slower execution times. Using shorter word lengths (5-7 characters) results in more query recalls (and thus more nonrelevant entries), lower memory usage, and faster execution times. For CASHE, words were clipped to a maximum of 8 characters for the inverted index. The clipping was done by computer during the creation of the inverted index. In addition, numerals and special characters were stripped out of the internal index.

8.5.2.3 Stop Words: To avoid a situation in which hundreds or even thousands of "hits" would be returned for a full-text query, there is a small group of terms, called *stop words*, that are not indexed. Stop words

include single letters of the alphabet, short words that are extremely common in English (such as *the* or *at*), and some words that appear in many or most CASHE entries (e.g., because they occur in standard subheadings). Table 4 shows the list of stop words for CASHE.

Table 4. Query Stop Words

The following is a list of the characters and words (truncated to 8 characters) that stop a query from being performed:

a b c d e f g h i j k l m n o p q r s t u v w x y z 1 2 3 4 5 6 7 8 9 0

adj	conditio	figures	methods	results	the
all	constrai	for	of	shall	these
an	cref	from	on	should	this
and	crefs	general	or	studies	to
applicat	cross	handbook	other	table	variabil
are	descript	in	procedur	tables	was
as	experime	is	ref	terms	were
at	fig	it	referenc	test	when
be	figs	key	refs	than	where
by	figure	may	repeatab	that	which
					with

9. SYSTEM SOFTWARE

The previous chapter described the data structures on the CASHE CD-ROM. The CASHE system software, or "engine," is what transforms these data into an interactive user interface. It reads the data files and displays their information on screen; copies, prints, or exports the information; searches the files in response to user queries; supports user extensions such as notes and bookmarks; and so on.

The CASHE system software can be broadly categorized as a set of specialized modules that perform specific tasks (such as search and retrieval or the display of figure components) and the main engine, which coordinates calls to and information exchange among the specialized modules, and carries out general system functions. This chapter focuses primarily on the specialized modules of the CASHE software, as listed in Table 5. The discussion is general, not technical, and is intended to provide a feel for how the software interprets and operates on the data structures described in the previous chapter to present information to the user and support the user's interactions with it.

Table 5. Primary Specialized Software Modules

Information Viewers
TextViewer
TableViewer
FigureViewer
OutlineViewer
GeneralViewer
Entry Palette
Retrieval module
CASHE Server
Session Manager

9.1 Information Viewers

Users interact with CASHE primarily through the information viewers (TextViewer, FigureViewer, TableViewer, OutlineViewer, and GeneralViewer). The interface for an information viewer is a specialized window type that understands and displays a specific class of information structured in an appropriate way. All viewer windows have:

- a drag bar (title bar), which allows the user to reposition the window;
- a banner (window title);
- a close box that can be used to close the window;
- a vertical scroll bar to reposition the information in the window;
- a size box that allows the user to change the size of the window;
- a zoom box that allows the user to toggle between the standard window size and a user-defined size;
- a content region where the data (text, figure, etc.) are displayed.

Some viewers also contain

- a control area;
- a horizontal scroll bar.

The functionality of the viewers is described in Chapters 3 and 4. (These chapters also provide illustrations of the interface for each viewer.) Here, we provide a more structural description that focuses on how each viewer interprets and displays the data in a file of its specific data type.

9.1.1 TextViewer

The TextViewer module is invoked to present the text components of the database documents. The responsibilities of the TextViewer include:

- Reading text from a text component file given a unique text file pointer.

- Displaying the entry text (along with any embedded graphics) in a vertically scrollable window in accordance with the appropriate style sheet.
- Maintaining the entry identification region at the top of the window, which contains context information.
- Detecting and handling mouse clicks on link markers embedded in the text (e.g., CRefs).
- Implementing string search of the text in the viewer window.
- Maintaining the left sidebar containing controls for navigation and user annotation, and handling mouse clicks on these controls.
- Recognizing selection of text with the mouse and supporting Clipboard operations.
- Supporting printing and saving operations.

When the system receives a request to open the text component of a database entry (for example, the user has selected the entry in an access outline or has clicked on a cross reference to it in another entry component), the system issues a call to the TextViewer and passes it a unique file pointer to the text component file that is to be opened. This component text file is a tagged ASCII file that contains both contextual information (e.g., banner for window title bar, entry number and title) and the text of the entry.

The viewer opens the text file, parsing its contents by reading and interpreting the SGML tags in the file. To render each tagged text element correctly on screen, the viewer consults an internal text style sheet that specifies the appropriate font, size, style, justification, etc., for each tagged item and also identifies any special characters required (such as Greek letters). Using the data in the text file and the specifications in the style sheet, the viewer opens a new window with the correct banner (which identifies the document, entry number, and component type), writes the entry number and complete entry title (and, for *EDC* entries, the topic area and subarea numbers and titles) into the title region at the top of the viewer window, and fills the content region of the viewer window with text.

Tags in the text file identify locations where PICT resources (e.g., mathematical formulas or in-line graphics) are to be inserted and specify the particular resource to be used. The viewer extracts these resources from the resources fork of the text component file and displays them in the appropriate locations.

Normally, the viewer flows text into the content region beginning at the top of the entry text. Sometimes, however, the TextViewer will be passed a file offset as well as a file pointer. In such a case, the TextViewer scrolls the window to the line indicated by the offset. For example, when the user selects a text component from a Query "hit" list, the system passes to the TextViewer the offsets of the beginning and ending of the query string, as well as the text file pointer. The TextViewer flows the text in the file into the content region of the viewer window, then scrolls the text to the line containing the query term and highlights the term.

The TextViewer handles window management—scrolling, dragging, or resizing the active window and activating and deactivating open TextViewer windows in response to user mouse manipulations. As described in sect. 3.1, only the text in the content region is scrolled when the user clicks the vertical scroll box—the elements in the entry identification region at the top of the window and the left sidebar are not scrolled but remain stationery. When the TextViewer window is resized by the user, the viewer reflows the text to fill the window at the new dimensions. The character (or graphic) in the upper left of the content region before resizing is defined as the first character. Text reflow occurs beginning with this character and continues until the content region is full. Because text is reflowed, horizontal scrolling is unnecessary and is not supported by the TextViewer.

The viewer detects when the cursor passes over an embedded (predefined) hyperlink, such as a figure or table call-out or a cross-reference to another entry, and changes the cursor to a pointing hand to alert the user to the presence of the hyperlink. All the information required to display and traverse the hyperlinks is contained in the tagged text files. Link markers (e.g.,

"CRef. 8.301") are simply tagged text embedded within the text contained in the TextViewer's field. The tag for each hyperlink identifies the type of hyperlink (e.g., to a figure or to entry text), provides the text for the link marker, and specifies a unique identifier for the destination component. Because all the pertinent information is contained in the markup, the viewer need not know ahead of time where the embedded hyperlinks are located. For example, when the viewer detects a mouse click on the phrase "CRef. 8.301," it scans that portion of text, extracts the hyperlink information contained in the tag, and initiates the hyperlinking process by passing the unique identifier for the hyperlink destination to the CASHE engine. The engine, in concert with the CASHE Server, then completes the execution of the hyperlink. Since the destination for the hyperlink in this example is a text component, the system passes back to the TextViewer the file pointer for the text component of entry 8.301. The TextViewer opens a new viewer window and fills it with the text of the component (or activates the appropriate window if the component is already open). If the hyperlink destination were, instead, a bibliographic reference (e.g., "Ref. 1"), the viewer would be passed an offset within the current file and would scroll the text to the location of the reference citation.

The TextViewer implements the Find function through a Find submodule. When a Find command is issued, the viewer passes the Find request to the Find submodule, which scans the text of the file displayed in the active viewer window for matches to the user's query string. This search is performed at run time. Find will search the text file for any string that can be typed in at the keyboard. If the query string is found, Find passes back to the viewer the offsets of the beginning and end of the string in the file. The viewer then scrolls the text in the window until the line containing the query string is at the top of the content region, and highlights the string.

The TextViewer supports the Clipboard. Entry text is selectable with the mouse by dragging across text regions. Selected text may be copied and pasted to the Clipboard (it may not be cut, pasted, or cleared, since the documents are read-only).

The TextViewer also supports the Print and Save Text As functions on the menu bar. When the user selects one of these commands, the TextViewer passes to the engine the text contents of the file in the active viewer window. The viewer consults built-in style sheets for printing and export to determine which structural/contextual file elements to include with the entry text, as well as typographical style and layout, and export file format. The full document citation is also added whenever the entry text is printed or saved.

The left sidebar of the TextViewer window contains two sets of controls. The top set (Previous Text Subsection arrow, Next Text Subsection arrow, and Text Subsection Selector) supports navigation within the entry text in the content region of the viewer window. The lower set of controls (Bookmarks, Notes, and Links) invokes user annotation functions. The TextViewer detects mouse clicks on any of these buttons and initiates the appropriate action for that control.

The data required to implement these navigation and annotation controls are not contained in the text component file. When the text file is initially loaded, the TextViewer obtains the required data for the navigation controls from the CASHE Server, which looks up the information in the component index. When the user clicks on the Next or Previous arrow, the TextViewer passes the unique identifier for the corresponding component to the CASHE engine to initiate the linking process. When the user selects an item from the Text Subsection Selector menu, the TextViewer scrolls the text to the appropriate offset.

When the text component is loaded into the TextViewer, the viewer communicates with the Session Manager to retrieve information about the component's user annotation objects. It then draws the correct icon on the sidebar (Bookmark button with or without the letter "B," link button with or without the center link, Notes button with or without the letter "n") to signal to the user whether the given type of user extension exists for that component. Bookmarks, user links, and notes are maintained by the Session Manager within session files. When the TextViewer detects a user click on any of the

annotation buttons, it calls the Session Manager to initiate the appropriate dialog with the user.

9.1.2 TableViewer

The TableViewer module is invoked to display the table components of documents. The TableViewer operates very similarly to the TextViewer. Its responsibilities include:

- Reading tabular text from a table component file given a unique table file pointer.
- Displaying the table (along with any embedded graphics) in a scrollable window in accordance with the appropriate style sheet.
- Implementing special freeze-pane scrolling of the table body.
- Maintaining the entry identification region at the top of the window.
- Detecting and handling mouse clicks on hyperlinks (e.g., CRefs, links to audio demonstrations).
- Implementing string search of the text in the viewer window.
- Maintaining the left sidebar containing controls for navigation and user annotation and handling mouse clicks on these controls.
- Recognizing selection of text with the mouse and supporting Clipboard operations.
- Supporting Print and Save operations.

When the system receives a request to open the table component of a database entry, it issues a call to the TableViewer and passes it a unique pointer to (and sometimes also an offset in) the table component file to be opened. Table component files, like text components, are tagged ASCII text that contains all the information the TableViewer needs to create the desired appearance and activate embedded hyperlinks. By consulting the internal style sheet for tables and reading the markup that identifies the table elements and

defines the column/row organization of the table, the TableViewer is able to render the table in the appropriate style and format.

The table component is displayed in a scrollable window with an identifying banner. If the viewer has been passed a file offset as well as a filename, the table is scrolled to the specified location. As with the TextViewer, the TableViewer maintains an entry identification region at the top of the window and a control sidebar on the left.

Some tables have embedded formulas or graphics that are stored as PICT resources of the table component file. The TableViewer identifies these embedded graphics from their tags, extracts the resources, and displays them in the appropriate location.

The TableViewer manages open viewer windows, scrolling, dragging, or resizing the active window, or bringing a new table window to the front of the desktop in response to user mouse clicks. The viewer keeps track of the location of column headings and row labels in the active table and implements special freeze-pane scrolling in which column headings are kept stationary for vertical scrolling and row labels remain stationary for horizontal scrolling. Only the body of the table scrolls. The table title (as well as the entry identification region and left sidebar) remains fixed.

The markup in the table file contains all the information necessary to display and activate hyperlinks in the table. The TableViewer detects when the cursor passes over a hyperlink embedded in the table (such as a figure call-out, a cross-reference to another entry, or a call-out for an audio demonstration) and changes the cursor to a pointing hand to signal the presence of the hyperlink. When the viewer detects a mouse click on an embedded hyperlink, it activates the hyperlink by passing to the CASHE engine the unique identifier for the destination file, which is specified in the tag.

A few tables have multiple panels accessible through radio button controls at the top of the table. The content for all table panels is contained in the table file. The tagged table file also contains all the information required to

portray and implement the hyperlinks between panels, including the location of each hot spot, the style of the link markers (radio buttons), the text labels for the markers, and the destination for each hyperlink. The TableViewer reads and interprets the tagging to draw in the radio button controls and implement the hyperlinks among table panels.

Like the TextViewer, the TableViewer implements the Find function by calling a Find submodule to process the user's request. The Find submodule scans the text of the file in the active viewer window for matches to the query string passed to it by the viewer. If it locates the string, Find passes the file offset for the term back to the viewer, which scrolls the table to bring the query string to the top of the window and highlights it.

Some tables in the CASHE database are copyrighted; a small minority of these cannot be printed or exported to the Clipboard or a file. The tagged table component file contains flags indicating when print and/or copy permissions have been denied. When the table file is first loaded into the TableViewer, the viewer reads this information and passes it to the CASHE engine so the corresponding command lines in the File and Edit menus can be inactivated if necessary. When printing and export are not blocked (the majority of cases), the TableViewer supports printing and saving to a file in the same way as does the TextViewer, consulting an internal style sheet to determine what data from the table file to include and how to format it, then passing this information to the CASHE engine on request. The TableViewer also supports copy of the table to the Clipboard. When the user invokes the Copy command, the entire table (not just a selected region of cells) is placed on the Clipboard. When a table has multiple panels, only the currently active panel is printed or exported. The entire active panel is output, including portions scrolled out of view.

The TableViewer maintains the navigation controls at the top of the left sidebar (Next Table arrow, Previous Table arrow, and Table Selector button) as well as the annotation controls at the bottom (Bookmark, Links, and Notes buttons). It handles mouse clicks to these controls in the same way as does the

TextViewer. When the table file is first loaded, the TableViewer obtains the required data for the navigation controls from the CASHE Server, which looks up the information in the component index. In response to user clicks on the Next/Previous arrows or selections from the Table Selector menu, the TableViewer passes the unique identifier for the corresponding table component to the CASHE engine to initiate the linking process.

The TableViewer also communicates with the Sessions Manager when the table component file is first loaded to learn whether or not user annotations exist for the component so the correct Bookmarks, Notes, and Links icons can be drawn on the sidebar. When the TableViewer detects a user click on an annotation button, it calls the Session Manager to process the user's request.

9.1.3 FigureViewer

The FigureViewer module is invoked to display the figure components of document entries. The responsibilities of the FigureViewer include:

- Reading the graphic from a figure component file given a unique figure file pointer.
- Displaying the graphic in a scrollable window.
- Maintaining the entry identification region at the top of the window.
- Detecting and handling mouse clicks on embedded hyperlinks.
- Maintaining the left sidebar containing controls for navigation, caption display, zooming, DataDigitizer, and user annotation, and handling mouse clicks on these controls.
- Calling the CaptionViewer submodule to display the figure caption in a separate window.
- Implementing string search of the text in the caption window.
- Supporting Copy, Print, and Save operations.

Each figure in the database documents is stored as a separate file, so that any individual graphic can be called via a unique file pointer. When the system receives a request to display a figure, it calls the FigureViewer and passes it the appropriate file pointer so the figure component file can be opened.

Figure component files are commented PICT files. The figure file contains the content for each graphic image that must be drawn in the viewer window. If the figure has several panels or a set of overlays, all these panels and overlays are stored in the same PICT file. In addition to the graphic image information, the file also contains contextual information (e.g., entry number and title), the full content of the caption and credit line, and permissions flags. This nongraphic information is in the form of tagged ASCII (as used for text and table components). The figure component file also identifies the default panel and overlay(s), and the location and link marker style of all embedded hyperlinks.

The FigureViewer parses the figure component file and consults an internal figure style sheet that specifies the appearance of nongraphic items such as the entry title and the link markers. The viewer then: (1) opens a window at the default size with the appropriate identifying banner in the drag bar; (2) writes the title and other contextual information into the identification region at the top of the window; (3) ascertains which is the default panel and overlay(s), locates the corresponding images in the file, and displays them in the content area of the viewer window; and (4) draws in any necessary link markers ("hot spots") that are not part of the artwork (such as radio buttons and check boxes).

The FigureViewer manages open viewer windows, moving, resizing, and activating or deactivating them in response to user commands. It scrolls the graphic in the active window vertically or horizontally when the user clicks the scroll bars at the right or bottom of the FigureViewer window. As with the other viewers, only the content region (that is, the graphic itself) is scrolled.

The entry identification region at the top and the control sidebar remain stationary.

The FigureViewer detects a mouse click on an embedded hyperlink and performs the necessary action. Most hyperlinks in figure components are calls to change panels or to add or remove overlays. When the viewer detects a mouse click on such a hot spot, it determines the required action, locates the destination panel or overlay, and redraws the content region in the necessary manner. For hyperlinks whose destination is outside of the current figure (such as a hyperlink to an audio demonstration), the viewer initiates the linking process by passing the unique identifier for the destination file to the CASHE engine.

The FigureViewer maintains the left sidebar and handles clicks to the navigation and annotation controls in the same way as does the TableViewer, communicating with the CASHE Server and Session Manager to set and implement the controls. The FigureViewer sidebar also contains three unique controls: the zoom buttons, the DataDigitizer button, and the caption button.

When the user clicks the Zoom In or Zoom Out button, the FigureViewer changes the cursor to a magnifying glass, supports the user in selecting a focus point for the zoom, and then, when the user clicks the mouse button, redraws the graphic into the content area, resizing it as required and placing the focus point in the center of the content area.

When the FigureViewer detects a user click on the DataDigitizer button, it issues a request to the CASHE engine to launch the DataDigitizer application, passing as input the graphic image in the content region of the viewer window.

When user clicks on the caption button are detected, the FigureViewer issues a call to the CaptionViewer. The CaptionViewer is a submodule of the FigureViewer that handles only caption display. The CaptionViewer operates similarly to the TextViewer, but with much reduced

functionality. When called, the CaptionViewer opens a plain window with an identifying banner constructed from the figure component file and flows into it the caption text contained in the file, consulting an internal style sheet to determine how to render the tagged text. The default caption window is relatively small so it can be opened on top of the figure itself without obscuring the graphic completely. The window is moveable and resizable, but has no entry identification region or control sidebar. The CaptionViewer maintains a vertical scroll bar, but reflows caption text into the window when the user changes the width of the window, so that horizontal scrolling is not necessary. Because the CaptionViewer is a submodule of the FigureViewer, the caption window is closed automatically when the associated FigureViewer window is closed.

The FigureViewer implements the Find function in the same way as do the other viewers. For FigureViewer, however, only the caption text is searched. When a Find command is issued, the CaptionViewer is called and the caption window is opened (or brought to the front of the desktop). The Find submodule scans the caption text for the user's query string and, if it is found, returns the offset to the viewer, which scrolls the window to the term.

The FigureViewer supports copy to the Clipboard, printing, and saving to a file in the same way as does the TableViewer. For figures with multiple panels and/or overlays, only the graphic image currently in view is printed or exported. The FigureViewer consults an internal style sheet to determine the content, style, and format of the figure file data to be printed or exported and passes these data to the CASHE engine on request. The caption, along with any accompanying credit line, is always included whenever the figure is printed or exported. As with tables, many figures are copyrighted, and some copyright holders denied permission to print or export the figure from within CASHE. This information is contained in the figure file and is read by the FigureViewer when the figure component file is first loaded so that the corresponding menu commands can be blocked.

9.1.4 Outline Viewer

The OutlineViewer module is used to display hierarchical structures such as the Integrated Outline and the tables of contents and back-of-the-book indexes for documents in the database. These outlines link to database components and are used to identify and access entries of interest to the user. The OutlineViewer is also used to display the outlines that provide a point of entry into the Glossary and Design Checklist. The OutlineViewer has the following responsibilities:

- Reading outline text from an outline file given a unique file pointer and constructing the corresponding tree data structure in memory.
- Supporting dynamic display of the outline (expansion and collapse of heading levels) in a vertically scrollable window in accordance with the appropriate style sheet.
- Detecting and handling mouse clicks on hyperlinks to external components.
- Implementing string search of the outline text in the viewer window.

When the user selects any outline item in the Documents menu, the system calls the OutlineViewer and passes it a unique file pointer to the outline file that is to be opened. Outline files are tagged ASCII text that use a "tree" and "leaf" markup system to specify the hierarchical relation between text elements. The OutlineViewer reads the outline file and interprets the tagging in the file to construct a tree data structure in memory. It consults an internal style sheet for outlines to determine how to render the outline headings and hyperlink structures on screen. It then opens a window at the default size, writes in the appropriate window banner, and flows the outline text into the window content area.

The OutlineViewer display mimics a printed outline. Subsections are embedded within (and indented under) sections in a hierarchical manner. The OutlineViewer initially displays only the highest level of the outline hierarchy for the given outline file. For the first-level file of a topically organized

structure such as a document table of contents, this will be a list of major sections; for an alphabetically organized outline such as an index or glossary, it will be the letters A-Z.

The viewer supports the user in browsing the outline by exposing or hiding a section's subsections in response to user mouse clicks. As specified by the style sheet for outlines, the OutlineViewer uses outline triangles similar to those employed by the Macintosh operating system in directory listings to represent the expansion state of a given outline heading. The viewer draws a right-facing triangle at the beginning of a heading to indicate that its subheadings are currently hidden and can be expanded. It draws a downward-facing triangle to indicate that the heading has already been expanded; the heading may be collapsed, but cannot be expanded further. The viewer detects mouse clicks on the triangles, expands or collapses the headings accordingly, and reorients the triangle appropriately. Heading expansion is stepwise; that is, expanding a heading displays only items at the next immediate level (i.e., only the heading's children, not its children's children, etc.) The scope of a "collapse" operation is "everything lower in the outline hierarchy"; that is, closing an expanded parent heading closes both its children and any children of its children, and so on.

The deepest level of the outline is the anchor level. Headings at this anchor level are hyperlinks to other documents. The OutlineViewer renders these headings in bold to differentiate them and precedes them by a filled rightward-facing triangle. The viewer recognizes when the user has positioned the cursor in this anchor line and changes the cursor from an arrow to a pointing hand to reinforce to the user that the heading is a hyperlink whose activation will change the user's context. When the viewer detects a mouse click on the anchor level, it reads the linking information embedded in the tagging for that heading and initiates the hyperlink by passing to the CASHE engine a unique identifier for the destination file. The destination for an outline hyperlink can be another outline file, an entry component file (text, table, or figure), or a non-entry text file (*EDC* Glossary or Design Checklist file).

As noted earlier, manipulating dynamic outlines in memory can become extremely slow and cumbersome when the outline files are large. For this reason, a two-stage process is used to expand lengthy access outlines like the document tables of contents or Integrated Outline that hyperlink to entry components at their lowest, anchor level. When the user selects such an access outline from the Documents menu, the Outline Viewer opens a file containing only the top two or three levels of the hierarchy. When the user clicks on a heading at the lowest level of this file, the viewer activates a hyperlink to a subordinate outline file by passing to the CASHE Server (via the engine) a unique identifier for the topic area heading the user has chosen to expand. The Server looks up the identifier in the component index and passes back to the OutlineViewer the pointer to the outline file containing the full expansion of that heading as well as the offset of the heading in the file. The OutlineViewer opens this subordinate outline in a new window and scrolls the window to the heading the user wishes to expand. Headings at the bottom level of this second outline file link directly to entry components.

The OutlineViewer manages open viewer windows in the same way as other viewers, scrolling, resizing, moving and activating/deactivating windows in response to user mouse clicks. The viewer supports vertical scrolling of window contents but reflows outline text to match the new window size when the user resizes the window horizontally.

As with the other viewers, the OutlineViewer implements string search of the text in the active window through a Find submodule that searches the outline file and returns a file offset to the viewer if the string is found. The viewer then scrolls the text to the appropriate location and highlights the search term. The entire outline file is searched. If the heading containing the string is currently collapsed, the viewer expands the necessary heading levels to display the line containing the search term.

The OutlineViewer does not support Copy, Print, or Save operations.

9.1.5 GeneralViewer

The GeneralViewer module is invoked to display supplementary or explanatory text information (and its embedded graphics), such as Glossary items, Design Checklist questions, and table pop-ups. The responsibilities of the GeneralViewer are:

- Reading text from a tagged text file given a unique text pointer.
- Displaying text (along with any embedded graphics) in a scrollable window in accordance with the appropriate style sheet.
- Detecting and handling mouse clicks on embedded hyperlinks.

The GeneralViewer is called by the CASHE engine when it receives a request to display a general text document that is not a database entry component. The engine passes to the GeneralViewer a pointer to the file that is to be opened and, when relevant, an offset designating the location in the file of the material to be displayed. Files opened by the GeneralViewer are tagged ASCII text. The viewer reads the tagged file, consults an internal style sheet for general text to determine how to render the material on screen, opens a scrollable window with the specified window banner, and displays the text appropriately in the window. If a portion of the Glossary or the Design Checklist questions is being displayed, the viewer scrolls the file to the offset passed to it by the engine so that the appropriate Glossary definition or Checklist question begins at the top of the window content region.

The GeneralViewer is often called in response to a hyperlink in an entry component that requests the display of supplementary material such as a Glossary definition or table pop-up that helps users interpret the material in the component. For this reason, the default size for GeneralViewer windows is only about one-fourth as large as that of component viewer windows, so that the underlying material is not totally obscured.

The GeneralViewer operates similarly to the TextViewer in its window management, text scrolling and text reflowing. However, the

GeneralViewer does not maintain an identification area at the top of the content region or a control sidebar.

The GeneralViewer displays table pop-ups that have embedded graphics stored as PICT resources of the text files. Markup within the file identifies the appropriate resource. When such a pop-up file is opened, the GeneralViewer identifies the embedded graphic from its tagging, extracts the graphic resource, and displays it in the content region.

As with other viewers, the GeneralViewer detects cursor movement over an embedded hyperlink and changes the cursor to a pointing hand. It responds to mouse clicks on the hot spot by reading the hyperlink tag contents and passing to the CASHE engine a unique identifier for the hyperlink destination. A hyperlink embedded in text displayed in the GeneralViewer can have as its destination: (1) another location in the same file (as may happen, for example, when one Glossary term contains a cross reference to another term); (2) a location in another general text file; (3) an entry component file; or (4) a QuickTime file containing an audio or video demonstration.

The GeneralViewer does not support string search of the text in viewer windows, nor does it support Copy, Print, or Save operations.

9.2 Entry Palette

The Entry Palette is a utility window that aggregates all the components of a given entry and allows the user to move easily and rapidly among them. The palette is a composite that serves to order the information in the database at a more coarse-grained level than the component level at which the text, figure, and table data are stored and displayed. It always reflects the entry associated with the component in the active component-viewer window.

The Entry Palette maintains a set of controls with pop-up menus that list the text component, figure components, table components, and test benches related to the given entry and allow direct navigation to any component or test bench. It also maintains controls that allow users to browse to neighboring

entries in the document. The active database entry is identified by document and number near the top of the palette.

The Entry Palette floats on top of all component windows. It is "co-active"; that is, controls on the palette can be used by clicking them without first clicking the palette to make it active, and clicking controls on the palette does not de-activate the topmost document window.

To implement its controls, the Entry Palette must have knowledge about: (1) the document and entry to which the component in the active window belongs and the identification number of the entry; (2) all the text, figure, and table components associated with the given database entry and the short title of each component; (3) all the test benches related to the entry and which topic (if any) should be flagged when the test bench is opened from the palette; and (4) the next entry and previous entry in the document. The palette obtains this information from the CASHE Server, which looks it up in the component index and passes the required data back to the Entry Palette. Because the content of the Entry Palette must always relate to the active component, the palette requests and receives the required entry data from the Server whenever a new entry component is opened or a different document window is brought to the front of the desktop, so it can update its control menus and entry identification line.

Hyperlinks from palette menus and buttons to entry components and test benches are handled in the same way as hyperlinks within viewers. When a Next Entry/Previous Entry button is clicked or a control pop-up menu item is selected, the Entry Palette initiates the linking process by passing to the CASHE engine a unique identifier for the requested component or test bench.

The Entry Palette does not support printing, copy/export, or text string search capabilities, since these functions are provided independently by each of the component viewers and test benches. Similarly, the palette does not handle annotation calls, since annotation functions are not available at the entry level, only at the component level.

9.3 Retrieval Module

The retrieval module supports full-text search of all documents in the database. The responsibilities of the retrieval module include:

- Responding to Query and related commands issued by the user via the Search menu.
- Managing the searching interface, such as the Query dialog box (including the term entry field, query history, and hit list display).
- Transforming the query entered by the user into a form acceptable for indexed retrieval.
- Performing indexed retrieval by accessing the index file and implementing required query operators (i.e., AND, OR, ADJ).

The retrieval module executes a Query command by consulting an inverted index of the text of all database documents that is stored on the CD-ROM (or on the user's hard disk). The inverted index is a list of all the words occurring in the database, clipped to a maximum of 8 characters, along with the locations in the database documents where each word is found. The only terms not included in the inverted index are words like *the* and *and* that are extremely common in English, and words that appear in standard *EDC* entry headings and thus are contained in almost every *EDC* entry. These "stop words" are omitted to avoid large and unmanageable hit lists that are of no value to the user. (A description of the inverted index and how it was created and a list of all stop words can be found in sect. 8.5.2.)

The user invokes full-text retrieval by means of the Query command in the Search menu. When a Query command is issued, the retrieval module opens the Query dialog box. This dialog accepts from the user a query string that may contain words or letters and a prescribed set of operators (AND, OR, ADJ). The retrieval module transforms the user's query string into an unambiguous and precise query expression that can be used in searching the inverted index. Only alphabetic characters (and parentheses for grouping) are

allowed in user queries. If the user types any nonalphabetic symbol into the query field (e.g., a numeral, a hyphen), the retrieval module deletes the symbol and replaces it with a blank space, which is echoed to the user in the query field (for example, "self-motion" is rewritten as "self motion"). When an operator (AND, OR, ADJ) is detected, the operator is echoed to the user in capital letters to distinguish it from a text string.

The retrieval module parses the query string, ignoring capitalization and truncating each word of the user's query to eight letters. Operators are executed in order of precedence, with ADJ operations completed first, then AND, then OR. Blank spaces between terms are interpreted as equivalent to the ADJ operator and are processed accordingly. Punctuation is ignored in determining adjacency. For example, the query "color coding" will find occurrences of "...color; coding..." or "...color. Coding..." in the text. Parentheses are parsed as in mathematics to override the normal precedence and determine the order in which operators are to be executed. Table 6 shows a formal specification of the CASHE query language grammar.

Once the query is parsed, the retrieval module searches the inverted index file for a given term and reads the set of entry component pointers and offsets indicating where that term is located. Query operators act as set operators on the sets of entry component pointers retrieved from the inverted index. Once the final list of pointers is compiled, the retrieval module communicates with the Server to obtain the information necessary to write the component titles list ("hit" list) in the Query dialog box. When the user chooses an entry component in the "hit" list, the retrieval module passes the corresponding pointer and offset to the CASHE engine so the component can be opened and scrolled to the first occurrence of the term.

The retrieval module keeps track of the user's 8 most recently used query strings, displays them at the user's request in a pop-up menu in the Query dialog box, and, if a string is selected from this menu, writes it in the query field of the dialog box.

The retrieval module also keeps track of the user's current selection in the hit list. It highlights this item in the list when an inactive Query window is reactivated. It also uses this information to execute the Previous and Next Match and Previous and Next Component commands in the Search menu on the menu bar.

Table 6. CASHE Query Language

Below is a formal specification, or grammar, of the CASHE search and retrieval query language. The grammar is presented in Backus-Naur Form (BNF). The notations '::=' and '|' are meta-symbols which mean "may be reduced to" and "or," respectively. Non-terminal symbols are represented in italic. Terminal symbols are represented in boldface. White space within productions is ignored and is only used for presentation purposes; literal spaces are represented as ' '. Long lists of items of the form "w|x|y|z" may be given in brackets: "[wxyz]." Notes that are not a part of the grammar are represented within braces {}.

<table>
<tr><td>expression</td><td>::=</td><td>term</td></tr>
<tr><td></td><td></td><td>| term or expression</td></tr>
<tr><td>term</td><td>::=</td><td>phrase</td></tr>
<tr><td></td><td></td><td>| phrase and term</td></tr>
<tr><td>phrase</td><td>::=</td><td>compound</td></tr>
<tr><td></td><td></td><td>| (expression)</td></tr>
<tr><td>compound</td><td>::=</td><td>word</td></tr>
<tr><td></td><td></td><td>| word adj compound</td></tr>
<tr><td></td><td></td><td>| word ' ' compound {implied adjacency}</td></tr>
<tr><td>word*</td><td>::=</td><td>letter</td></tr>
<tr><td></td><td></td><td>| letter word</td></tr>
<tr><td>letter</td><td>::=</td><td>[ABCDEFGHIJKLMNOPQRSTUVWXYZ</td></tr>
<tr><td></td><td></td><td>abcdefghijklmnopqrstuvwxyz]</td></tr>
</table>

*Words are not allowed to be equal to the reserved words AND, OR, and ADJ, which are operators, and thus terminal symbols.
Words preserve their sense across all capitalization schemes. For example: "adj," "ADJ," and "aDj" are all treated as the same word.

9.4 CASHE Server

The CASHE Server resolves all data requests generated in the CASHE application. Data requests are issued, for example, when the user selects a menu item that hyperlinks to an access outline or entry component, chooses an entry component from the search "hit" list in the query dialog box (or from a list of components in one of the annotation dialog boxes), or clicks on an embedded hyperlink in an outline, entry component, or other file.

All such hyperlinks must be referred to the component index, which is the single source of knowledge about entry components, other data elements, and the locations of data files on the disk (see sect. 8.5.1). The Server is the only module that interfaces to the component index. The hyperlink information contained in a menu item or embedded in a file at the location of a link marker is only an identification label for the hyperlink destination. For the hyperlink to be fully processed, this identifier must be looked up in the component index to find the corresponding file pointer (or pathname) so the destination component can be opened. The Server is the module that handles these requests.

For example, if the user clicks on a "Figure 1" hyperlink in the TextViewer, the viewer sends the hyperlink identifier to the CASHE engine, which passes it to the Server. The Server looks up the identifier in the component index, interprets it to be a request for a figure component, reads the filename for the figure, and passes this information back to the CASHE engine. The engine then calls the FigureViewer and passes it the figure filename, so the figure can be opened.

Data requests for non-entry data items are processed similarly, For instance, if the Server receives a data request for a Glossary term, it looks up the term in the component index, interprets it to be a request for a general document (as opposed to a component text document), and ascertains which of the 11 Glossary data files contains the term and the offset of the term within the file. Then the Server passes this information back to the CASHE engine so

the correct viewer can be called and the file can be opened and scrolled to the term.

The Server module is stored as a separate application on the CD-ROM. Nevertheless, the Server is an integral part of the CASHE software that communicates continuously with the engine to facilitate data retrieval.

9.5 Session Manager

The Session Manager maintains the session objects that are created when the user runs the CASHE software. The Session Manager has the following responsibilities:

- Managing the annotations interface (including the Bookmarks, Notes, and Links dialog boxes for creating and editing annotations).
- Maintaining the set of annotation objects (notes, hyperlinks, and bookmarks) created by the user.
- Maintaining the history list (a list of the entries visited by the user during a given session).
- When requested by the user, saving all session objects to disk in session files and reading and restoring these objects when session files are reloaded.

When the user decides to create or edit an annotation for a given entry component and clicks one of the annotation buttons on the sidebar of a component viewer, the viewer issues a call to the Session Manager. The Session Manager then displays the appropriate dialog box and receives and processes user input.

For Bookmarks and Notes, the Session Manager displays a dialog box containing a title entry field into which the user can type a name for the bookmark or note, if desired. For Notes, the dialog box also contains a text field for entering the content of the note. The Session Manager supports rudimentary text-editing operations for the notes field. Users can create text

notes that contain any character in the Helvetica typeface. Font changes, custom spacing, and special effects such as boldface or underlining are not supported. Users can employ the dialog boxes to create new bookmarks or notes or to edit or delete existing bookmarks and notes.

The Session Manager manages hyperlink creation and navigation through two concatenated dialogs. When the Links button on the viewer sidebar is clicked, the Session Manager displays a Link Destinations dialog box listing the destination components for all user-created hyperlinks attached to the current component. The Session Manager supports direct navigation to any component on the destination list. When the user requests activation of a hyperlink in this list, the Session Manager passes a unique component identifier for the selected destination to the CASHE engine, which initiates the process of opening the destination component. Controls in the dialog box also allow the user to remove existing hyperlinks or create new hyperlinks. When the user clicks the New button in the dialog box, the Session Manager displays a second dialog that allows the user to set a new hyperlink and name the hyperlink, if desired.

The Session Manager keeps track of all the user annotations created during the current session. Whenever a new component is opened, the viewer that opened it queries the Session Manager regarding the existence of annotations for that component so that the correct annotations icons can be displayed.

The Session Manager maintains lists of all notes, bookmarks, and user-created hyperlinks for the session, which can be accessed by the user through the Annotations menu on the menu bar. Selecting Notes, Links, or Bookmarks from the menu results in a call to the Session Manager, which displays a Session Notes, Session Links Source, or Session Bookmarks dialog box listing all annotations of the corresponding type for the current session. For example, the Session Bookmarks dialog box lists all components on which the user has placed bookmarks, along with the title the user has given to each bookmark. The Session Manager supports direct navigation to entry

components listed in the dialogs. If the user selects a component in the bookmark list, for instance, the Session Manager passes a unique component identifier for the selected component to the CASHE engine, which, in communication with the Server, locates the component file and calls the appropriate viewer to open it.

The Session Manager also maintains a list of the last 12 text components that have been visited during the current session. This history list is displayed when the user pulls down the History menu on the main menu bar. When an item in the menu is selected, the Session Manager issues a call to the engine and passes it an identifier for the selected component, so the component can be opened (or brought to the front of the desktop if it is already open).

When the user requests that a session be saved, the Session Manager assembles the annotation objects (user notes, bookmarks, and hyperlinks) and the history list. It also queries the engine for the desktop layout of entries, including a list of identifiers for all open components as well as the position and size of the viewer window in which each component is displayed. The Session Manager then formats all this information (annotations, history list, and desktop layout) and stores it to disk under a filename selected by the user. When the user loads a previously saved session, the Session Manager restores all these objects back into the CASHE environment, passing to the engine the requisite file identifiers and window information so the preexisting CASHE desktop layout can be re-created.

9.6 Residual Engine Functions

The main CASHE engine coordinates calls to the specialized modules (such as the viewers and the Server) and the exchange of information among them, interfaces with the Macintosh operating system as necessary, and handles all residual functions not assigned to one of the specialized modules.

A few of these functions are described very briefly below, but no attempt is made to catalog them here.

The engine oversees all data requests and routes information to and from other modules to see that such requests are implemented properly. For example, the engine might receive a request from the TextViewer to open a figure component. It passes the request to the Server and receives back a file pointer and identification of the document type (figure). It checks with the Macintosh operating system to see if the component is already open on the desktop. If so, it brings the window containing the component to the front of the desktop. If not, it issues a call to the FigureViewer and passes it the figure file pointer. The FigureViewer then opens the file in a new FigureViewer window.

When requests from component viewers involve external applications, the engine issues appropriate calls. For example, it calls the Apple MoviePlayer to play CASHE QuickTime files when users select demonstration hyperlinks in component viewers. It also issues calls to open external CASHE applications such as the test benches, DataDigitizer, and on-line Help application.

The engine receives and handles input/output requests and other menu bar commands, routing requests to the appropriate module, obtaining required data from other modules, and interfacing with the operating system. When the Print command is selected, for example, the engine requests the formatted data from the active component viewer, issues the appropriate system calls, and passes the formatted data to the Macintosh Print Manager for printing.

As with other Macintosh applications, desktop accessories and other items under the Apple menu are always available to the user while running CASHE. Users can switch back and forth at will between CASHE and other open applications on the desktop.

9.7 Software Testing

The custom-written software comprising the CASHE engine was tested thoroughly before being installed on the CD-ROM. Working from the system specification, user's manual, and other project documentation, an outside consultant drafted an exhaustive test plan that provided formal procedures for testing all elements of the user interface, including the menu bar items, Entry Palette, information viewers, user annotations, full-text search, component-level search, navigation lists, session support, context-sensitive help, and figure/table copyright protection. The software testing was conducted in house. Every available command was executed, every control activated, and every function invoked, to verify that the software operated correctly and that the correct displays were presented. For example, testers selected every menu item in the menu bar and clicked every sidebar control in every component viewer.

In addition to being tested extensively, the CASHE software was also evaluated by a Macintosh expert to make sure it conformed to Macintosh human interface guidelines.

10. INTEGRATED EXTERNAL APPLICATIONS

The CASHE CD-ROM contains several other software programs in addition to the main CASHE application and its associated server: the test benches in the Perception and Performance Prototyper, the DataDigitizer, and the on-line Help. Although these items are all completely integrated into CASHE through the user interface, they are separate applications and may be run independently of the CASHE engine.

10.1 Test Benches

Developing the software code for the 11 test benches in the Perception and Performance Prototyper was a process that echoed the development of CASHE in some ways, because it entailed the need to communicate information about human behavioral phenomena to team members for whom this was not the primary area of expertise. The content of each test bench — what phenomenon was to be simulated, which variables should be included, how the user should be able to interact with the phenomenon — was defined by a subject-matter expert familiar with the test bench topic. The task was to communicate this content to software engineers, who had no formal training in the perceptual and performance effects being simulated, so that proper presentation of the test bench effects would be assured. To accomplish this task, a test bench development process was devised that included a detailed technical specification of each test bench and several review cycles to verify that the test bench conformed to requirements. (See Fig. 39.)

10.1.1 Test Bench Topic Selection

The first step in test bench development was to select a set of test benches that would adequately cover the knowledge base. The entries in the

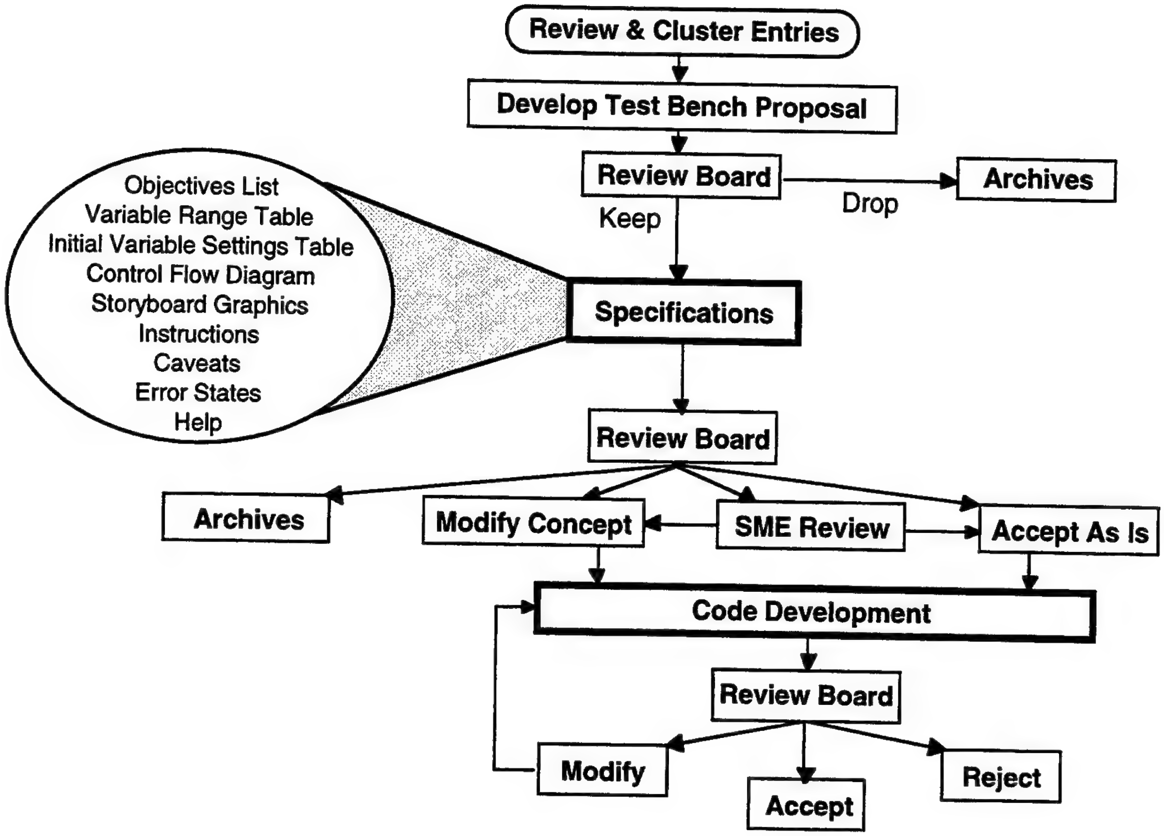


Figure 39. Test bench development process

database were surveyed, and entries were clustered to identify the common underlying behavioral concepts and select good candidates for test benches. Over 60 potential test bench topics were identified. These fell into the twelve broad subject areas of human perception and performance shown in Table 7. These subject areas are believed to be central to the design of complex human-machine systems.

An outline proposal was written for each of these 60+ candidate test benches defining the behavioral effects to be illustrated and the database entries on which the test bench was to be based. These proposed test benches were then reviewed to narrow the list to a subset that would be developed for version 1.0 of CASHE. A review board that included experimental psychologists, human

Table 7. Subject Areas of Potential Test Bench Topics

Visual performance
Spatial and temporal perception
Color
Auditory performance
Auditory grouping and localization
Language and display processing
Vigilance, monitoring, and search
Motor control
Reaction time/accuracy
Effects of environmental stressors
Memory
Methods and techniques

factors specialists, engineers, and programmers rated each proposed test bench according to a set of weighted criteria that focused primarily on their usefulness to system engineers, the fidelity and reliability with which the given phenomenon could be reproduced, and the technical feasibility of creating the test bench for the Macintosh platform. These ratings were then used to select the set of test benches to be developed. Specifications were written for about one-third of the identified test benches. A second review of the specifications produced a further reduced set for immediate implementation and prioritized the remaining test benches for future development.

10.1.2 Development of Specifications[1]

Full specifications were written for the 11 test benches programmed for CASHE version 1.0. The outline proposal drafted for each test bench during

[1] For more information on the development of specifications for these interactive documents, see "Developing Behavioral Phenomena Test Benches," by D. L. Monk, S. J. Swierenga, and J. E. Lincoln, 1992, *Proceedings of the Human Factors Society 36th Annual Meeting*, *2*, pp. 1106-1109.

the database review served as the starting point and core definition for the full test bench specification. In addition, a library of components required for all test benches was compiled to aid test bench designers. Subject-matter experts, working from these materials, developed a detailed set of design specifications for each test bench that included the following items:

(a) An objectives list, which identified the phenomena, effects, and/or relationships among the variables that were to be experienced by users of the test bench.

(b) A table identifying the set of stimuli to be used, the stimulus characteristics or variables the user would be allowed to manipulate, and the range of variable values that would be supported.

(c) A definition of the general approach to be used (reproduction of effects, simulation of effects, user self-test, data access, as described in sect. 7.2.1).

(d) A control flow diagram identifying all possible user interactions with the test bench.

(e) Storyboard graphics describing what the user was to see and hear during each step of the control flow; these included layouts for each screen, designs for all screen objects (e.g., controls, text blocks, graphics, and audio), and complete descriptions of visual and auditory effects for each display screen in the control flow.

(f) A table defining how the test bench and its individual modules were to be linked to entries in the database and to other test benches, including a specification of the default state of the test bench when entered from each linked database entry.

Each completed set of specifications was edited and reviewed by another subject-matter expert, a human factors specialist, and a software engineer to check its conformity to the original proposal, confirm technical feasibility, and assure consistency with other test benches in general design

and operation. Any confusion or uncertainty about what the test bench was intended to demonstrate and how it should operate was resolved and specifications were amended and clarified as necessary.

10.1.3 Test Bench Coding

The test benches were programmed in SuperCard, with XCMDs (external C routines) to control stimulus presentations and user data collection. To speed programming and promote consistency among the interfaces for the different test benches, a common shell was used for all test benches. The shell is responsible for:

- drawing in the main menu bar and handling selections from its menus;
- opening and maintaining the standard test bench viewer window template;
- recognizing user clicks on sidebar controls in the viewer window and executing the appropriate actions, such as linking to other test benches or to *EDC* entries that provide additional information about the test bench topic; displaying the test bench topic selection menu; and displaying information dialog boxes containing commentary about the test bench or instructions for operating it;
- storing the user's self-test results and displaying them when requested;
- displaying the current settings of stimulus and task parameters for individual test bench topics.

In addition to the shell, each individual test bench contains additional custom SuperCard code that creates the topic screens specific to that particular test bench, from which the user launches demonstrations and mini-experiments. The stimulus displays themselves, as well as the self-test event flows, were generally programmed in C and launched from within SuperCard as XCMDs (external commands). Speech stimuli and several other complex sounds used in some of the test benches were prerecorded using SoundEdit.

A general graphic design was developed for the test bench interface by a graphics studio. This overall style was used for all the test benches to provide a consistent look and feel to the display screens. Some of the required screen artwork was produced in house and some was drawn by a design studio.

10.1.4 Review and Testing

Once the test benches were coded, they were again reviewed by subject-matter experts and human factors specialists to verify that the demonstrations accurately produced the desired perceptual and performance effects. Each test bench was then converted into a separate, self-executing SuperCard application.

Once completed, each test bench underwent thorough software testing before being stored on the CD-ROM. Testing examined the standard functions available for all test benches to make sure they had been implemented appropriately for the given test bench. All menu bar functions were checked, the overall control flow of standard shell functions was verified, and the content of all cross-referencing menus and information dialog boxes was checked for accuracy. Each individual test bench topic was examined carefully to make sure that stimulus controls operated properly, that stimuli were displayed correctly, and that the timing, flow, and data collection of mini-experiments were implemented accurately. Test bench software was tested on a variety of Macintosh models to verify that the test benches operated properly on hardware systems with different CPU speeds, sound chips, monitors, etc.

10.2 On-line Help

CASHE on-line help, which reproduces the printed *User's Guide* (except for the artwork), was programmed as a simple stand-alone SuperCard application. To make the guide easier to use on-line, its contents were divided into small topical chunks based on the natural divisions of headings and subheadings. Each topical chunk was placed on a different SuperCard “card”

and is displayed separately. Most chunks fit within the display window so the material on each topic can be read without scrolling. The front end of the on-line guide is a multilevel hierarchical browser created from the Table of Contents of the printed guide. It operates like the other CASHE outline browsers to enable users to locate a specific topic of interest. The bottom level of the hierarchy links to the "card" containing information on that topic.

The electronic version of the *User's Guide* prepared during typesetting of the printed version was used to create the on-line help stack. Programming was done in house. The completed SuperCard stack was tested in house to verify content and control flow.

10.3 DataDigitizer

The third stand-alone application linked integrally with CASHE is the DataDigitizer, an application that allows users to digitize data graphs and save the data to a file for subsequent analysis and comparison. The DataDigitizer is custom software written by the outside contractor that supplied the CASHE engine software. DataDigitizer software was tested at the same time the engine software was tested.

11. PRODUCT RELEASE

11.1 Packaging

The CASHE CD-ROM is shipped in a jewel box case with insert. The package includes a printed 260-page *User's Guide* and a registration card.

11.1.1 Jewel-Box Insert

The cover and insert of the jewel box provide a quick overview of the product, including contents, major features, sponsors, and distributor. The insert also outlines the required system setup and provides brief instructions for installing the CD-ROM.

11.1.2 User's Guide

The CASHE interface is designed to be simple, straightforward, and intuitive to operate. However, a comprehensive 260-page *User's Guide* is provided with the CD-ROM for users who prefer to review the operation of the product before launching a session, who want to learn the interface faster, or who run into difficulties while running CASHE.

The primary goal of users in consulting the CASHE product is not to become an expert in the software, per se, but to locate behavioral data relevant to their specific needs as quickly as possible. The design of the *User's Guide* was approached from a human factors perspective to assure that it supported users in meeting this goal. Information in the guide is designed to be easy to find and use, comprehensible to novices yet detailed enough for expert users.

The guide provides information on system configuration, instructions for installing CASHE, and troubleshooting assistance, including a

list of error messages and brief suggestions to remedy the errors. A general overview of the product introduces users to the CASHE database and describes what documents are included, how to look for information efficiently in the database, and how the information viewers operate. Also included is a more detailed reference guide that explains the operation of all menu selections, viewer controls, etc.

The operation of each test bench in the Perception and Performance Prototyper is described fully. Technical aspects of visual and auditory display generation are also detailed to assist users in evaluating the meaning of test bench results in their specific design environments.

Development of the *User's Guide* entailed the same production steps as those required to publish any 260-page book: writing, editing, design and layout, figure drafting, typesetting, proofreading, preparation of the index, cover design, and printing and binding. The guide was spiral bound so that it will lay conveniently flat when used.

11.1.3 Release Notes

CASHE:PVS Version 1.0 Release Notes are one of the files copied to the user's hard disk during CASHE installation. The Release Notes provide complementary or late-breaking information as a supplement to the *User's Guide*. They can be read or printed using SimpleText (the text editor supplied with the Macintosh operating system) or any word-processing application.

The Release Notes contain the most up-to-date information on CASHE system requirements and installation. In addition, they explain how to access on-line Help, how to recognize entries with special audio demonstrations or animations, how to obtain the best performance in running the test benches, and how to obtain technical support. The Release Notes also include changes and updates to the *User's Guide*.

11.1.4 User Registration

Each CASHE package includes a user registration card. User registration will allow project administrators to track the installed base of CASHE users. Registration will benefit users by permitting them to receive notification of upgrades or enhancements to the product.

11.2 Installation

CASHE is installed by inserting the CASHE CD-ROM into the CD-ROM drive, double-clicking its icon to display the disk contents, then double-clicking the CASHE:PVS 1.0 Installer icon to launch the installer.

Some read-only files must be copied from the CD-ROM to the user's hard disk. Rather than expend the considerable resources required to develop a dynamic multistaged CD-ROM caching system, files were selected during the design process for "static caching" on the hard disk. These are copied from the CD-ROM at installation time. To permit CASHE to run on a wider variety of systems, these files are grouped into two distinct sets: the first set includes files that must be installed on the hard disk from the CD-ROM; the second set includes additional files that may optionally be installed as well. Installing just the mandatory set will result in minimally acceptable performance on the entry-level platform; installing both sets will result in nominally acceptable performance on the entry-level platform.

The standard (full) installation copies both the required and optional sets of files to the user's hard drive. These files include the CASHE applications, indexes, CASHE DTDs and other SGML-related files, and Release Notes. The full installation requires 27 MB of disk space. This installation yields the best CASHE performance. Users who do not have enough free hard disk space can elect to perform a minimal installation that requires only about 4 MB of space. The minimal installation leaves the large indexing files on the CD-ROM and places only the applications and other required files on the hard drive. Although less storage space is required, CASHE performance will be

substantially slower when entry components are opened or text searches are performed. Either installation places QuickTime 2.0 and Thread Manager 1.1 in the system extension folder if they are not already present or if a lower version number of either exists.

After the user installs CASHE, the user's hard disk will have a new CASHE:PVS folder containing the CASHE:PVS application, CASHE:PVS Release Notes, a CASHE:PVS Files folder, the CASHE:PVS Help application, and DataDigitizer, as well as several data files and the CASHE:PVS Server application.

As with other Macintosh applications, CASHE can be launched by double-clicking the CASHE:PVS application icon. CASHE may also be launched by double-clicking any CASHE session files that have been saved previously.

The files and folders within the CASHE:PVS folder on the user's hard disk must not be deleted, moved, or renamed (although the CASHE:PVS folder itself may be renamed and placed wherever the user desires). However, users may make an alias of the CASHE:PVS application and place it elsewhere. CASHE session files created by the user may be placed anywhere on the user's disk.

When CASHE is running, the CASHE:PVS application will run another program, the CASHE:PVS Server. The server is invisible to users. Both the application and the server use system memory.

11.3 Customer Support

When a question arises or a problem occurs, users can consult balloon help, the on-line Help application, or the printed CASHE:PVS *User's Guide*. Additional technical support is available from customer support staff at the Crew System Ergonomics Information Analysis Center (CSERIAC), at Wright-Patterson AFB, OH. CSERIAC handles promotion, sales, and technical support for the CASHE:PVS CD-ROM.

12. FUTURE DIRECTIONS

12.1 Scale-Up

From the beginning, CASHE version 1.0 was designed as an R&D release, intended to demonstrate the feasibility of the basic approach and suggest a direction for future products. Scale-up is envisioned along three dimensions, in particular: increasing the number of documents in the database, increasing the number of visualization enhancements, and increasing the base of delivery platforms.

12.1.1 More Documents

CASHE was specifically designed as a multidocument database. One of its primary goals was to demonstrate how disparate documents can be incorporated within a standardized interface and linked in a meaningful way. The software was designed to provide some specific opportunities for expansion. The design is highly tuned to encyclopedic documents accessed via outline browsers; thus, new documents of this type will be the easiest to support.

A new document may be added to the system, first, by defining its entries (i.e., units) and their components (text, tables, and figures), and then specifying a scheme for referencing these components. Although the current TextViewer, TableViewer, and FigureViewer should be broadly adaptable, a new component viewer can be added if the new document contains components that cannot be easily displayed using the existing viewers.

New outline files (such as a Table of Contents) may be added to provide access to the new document. New menu items to invoke these outlines must also be added. As long as new outlines conform to the outline indentation

guidelines, they are compatible with the existing OutlineViewer. In addition, the lowest-level outline elements must be declared component pointers acceptable to the component viewers.

The entries of the new document will have to be mapped onto the Integrated Outline, which covers all documents in the database. For maximum integration and usability, cross-reference links between the new document and the existing database document will also need to be added. Although interdocument cross-referencing was done by hand in CASHE version 1.0, we are investigating ways to wholly or partly automate this task as more documents are added.

12.1.2 MIL-STD-1472D Reference Documents

MIL-STD-1472D cites 43 other standards and specifications. These citations are different from the references found in most other documents, however. Rather, the sections of these documents that are cited by *MIL-STD-1472D* are included in it by reference; that is, they become an integral part of *MIL-STD-1472D* for purposes of government regulations and contractor obligations.

In most instances where an external document is referenced in *MIL-STD-1472D*, an explicit subsection rather than the entire document is cited. In future versions of CASHE, it is planned to include the specified text and/or graphics from the referenced document as well as document identification information, appropriately linked to the *MIL-STD-1472D* citation. Thus, CASHE users will be able to view the external document subsection when reading the *MIL-STD-1472D* section that cites it.

12.1.3 More Test Benches

In addition to the 11 test benches programmed for CASHE version 1.0, over 50 other possible test bench topics have been identified. These candidate topics have been reviewed and prioritized by subject-matter experts, human

factors specialists, and programmers. Storyboards have even been written for a few of the topics. As time and funding permit, additional test benches can be added to the CD-ROM to increase the number of perceptual and performance phenomena that can be investigated by users.

Reviewers have also identified additional video and audio demonstrations that can profitably be added in the future to help increase users' understanding of behavioral concepts and effects discussed in the database.

12.1.4 More Platforms

Version 1.0 of CASHE was programmed for the Macintosh line of computers, primarily because of the superior multimedia capability available for this platform when the project was initiated. Future expansion of the CASHE CD-ROM could include extension of the product to run on IBM PCs and compatibles, on workstations, and on the World-Wide Web. In fact, discussions are already under way concerning development of a PC version of the product.

12.2 Future Enhancements

In addition to enlarging the database, we are considering a number of functional enhancements for future versions. Some are refinements that were impossible to implement in version 1.0 because of time and cost constraints. Others are features suggested by user feedback and our experience with the product.

12.2.1 System features

- Ability to narrow Query text searches to specific database documents and to specific entry fields within documents (for example, a user could search only the *EDC* document and only the "General Description" subsection of *EDC* entries)

- Expanded two-phase linking, in which clicking a link marker would identify the link destination (e.g., by displaying the title of the destination entry) before activating the link, allowing the user to evaluate the relevance of the link before actually following it
- Dual presentation of user-created notes both as annotations to individual components and as a cumulative, browsable Notes file
- Support of fine-grained user annotations (i.e., users could attach notes, bookmarks, or links to specific locations within a component, such as a specific table row, instead of just to the component as a whole)
- Support of graphics within user-created notes (only text is currently allowed in notes)
- Expansion of context-sensitive help
- Provision of a separate window for bibliographic references so users could view citations referred to in the text simultaneously with the text itself (currently, entries are scrolled to the appropriate citation in the Key References section when users click a "Ref." link)
- Addition of a cumulative author/reference list for database documents that would combine all bibliographic citations from all documents into a single alphabetic listing (with an entry for each author when a cited work has multiple authors)
- Addition of a combined Glossary covering all database documents

12.2.2 Help tools

- Addition of graphical section maps—schematic diagrams showing the relationship among the entries in a given section in the form of a "flow chart" linking related topics, which could be used to locate specific entries relevant to the user's needs (the printed *EDC* already contains such logic diagrams, but they were not implemented for CASHE version 1.0)
- Provision of access filters targeted to specific task areas and/or user

subgroups to make it easier for inexperienced users to find information relevant to their specific needs; such filters might take the form of specialized browsing outlines, mission tree diagrams, checklists, or other forms of access aids that organize the database content from a specific viewpoint

- Implementation of interactive formulas; this feature would allow users to insert a desired value into generalized formulas presented in the database documents (such as the formula for computing visual angle or the formula for determining decibel level); the system would then compute the value of the mathematical equation
- Implementation of interactive data graphs; this feature would allow users to click data points on some types of graphs and experience an approximation to the physical stimulus used to collect the data represented; for example, a user might click a data point on a graph of equal loudness contours and hear a tone of the appropriate frequency and sound level
- Provision on the CD-ROM of tutorials and/or guided tours of CASHE; such tutorials would demonstrate CASHE features and guide users in learning how to use the product to obtain information relevant to specific design problems and issues; CASHE tutorials are already available off line

12.2.3 User customization

- Accommodation of user preferences by allowing users, for example, to select among alternative style sheets for displaying material on screen, to set default window size, and to specify window tiling vs. stacking for multiple windows
- Support for users in creating customized hierarchical browsing outlines linked to specific database entries
- Support for printout of session objects (such as notes, bookmarks and user-created links)

- Provision of more extensive project files incorporating, for example, user-created outlines, query lists and search results lists, a cumulative Notes file, user data files from test bench mini-experiments and DataDigitizer analyses, and a system state file to restore complete system state
- Extension of import capabilities to more test benches, so users can import their own visual or auditory designs for tryout in the P^3 test benches

13. CASHE BIBLIOGRAPHY

Boff, K. R., and Monk, D. L. (1992). Computer-aided systems human engineering: a hypermedia tool. *CSERIAC Gateway, 3*(2), 1-5.

Boff, K. R., Monk, D. L., and Cody, W. J. (1991). *Computer aided systems human Engineering: A hypermedia tool.* Paper presented at RIAO 91, Intelligent Text and Image Handling, Barcelona, Spain, April 2-5.

Boff, K. R., Monk, D. L., Swierenga, S. J., Brown, C. E., and Cody, W. J. (1991). Computer-aided human factors for systems designers. *Proceedings of the Human Factors Society 35th Annual Meeting, 1*, 332-336.

Brown, A. S. (1993). CASHE lightens the data load. *Aerospace America, 31*(9), 36-37.

Cona, T. R., and Monk, D. L. (1993). Bringing human performance data to the design table. *Proceedings of the Human Factors and Ergonomics Society 75th Annual Meeting, 2*, 1087-1090.

Cona, T. R., and Monk, D. L. (1994). Improving the impact of human performance information on the process of human-system design. *Symposium on Human Interaction with Complex Systems,* 247-250.

Lincoln, J. E. (1995). The CASHE:PVS hypermedia ergonomics database for system designers. *CSERIAC Gateway, 6* (2), 11-12, 14.

Lincoln, J. E., Brown, R. D., and Monk, D. L. (1996). A Flexible Human-Computer Interface for Accessing and Interpreting Human Performance Data. *Symposium on Human Interaction with Complex Systems.*

Lincoln, J. E., and Monk, D. L. (1997). Displaying Scientific Graphics on Computer. *IEEE Transactions on Professional Communication, 40* (2), 78-91.

Matthews, B. (1996). Software Review—Computer Aided Systems Human Engineering: Performance Visualization System. *HFAC/ACE Communiqué*, *26* (3), 4-5.

Monk, D. L. (1994) Computer Aided Systems Human Engineering: Performance Visualization System (CASHE: PVS)—An interactive hypermedia design tool. *Proceedings of the Human Factors Society 38th Annual Meeting 2,* 948.

Monk, D. L., Swierenga, S. J., and Lincoln, J. E. (1992). Developing Behavioral Phenomena Test Benches. *Proceedings of the Human Factors Society 36th Annual Meeting*, *2*, 1106-1109.

Swierenga, S. J., Monk, D. L., and Brown, C. E. (1992). Human performance data visualization for system design teams. *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON), 2*, 675-680.

APPENDIX A

CONTROL FLOW DIAGRAMS FOR VIEWERS AND SELECTED FUNCTIONS

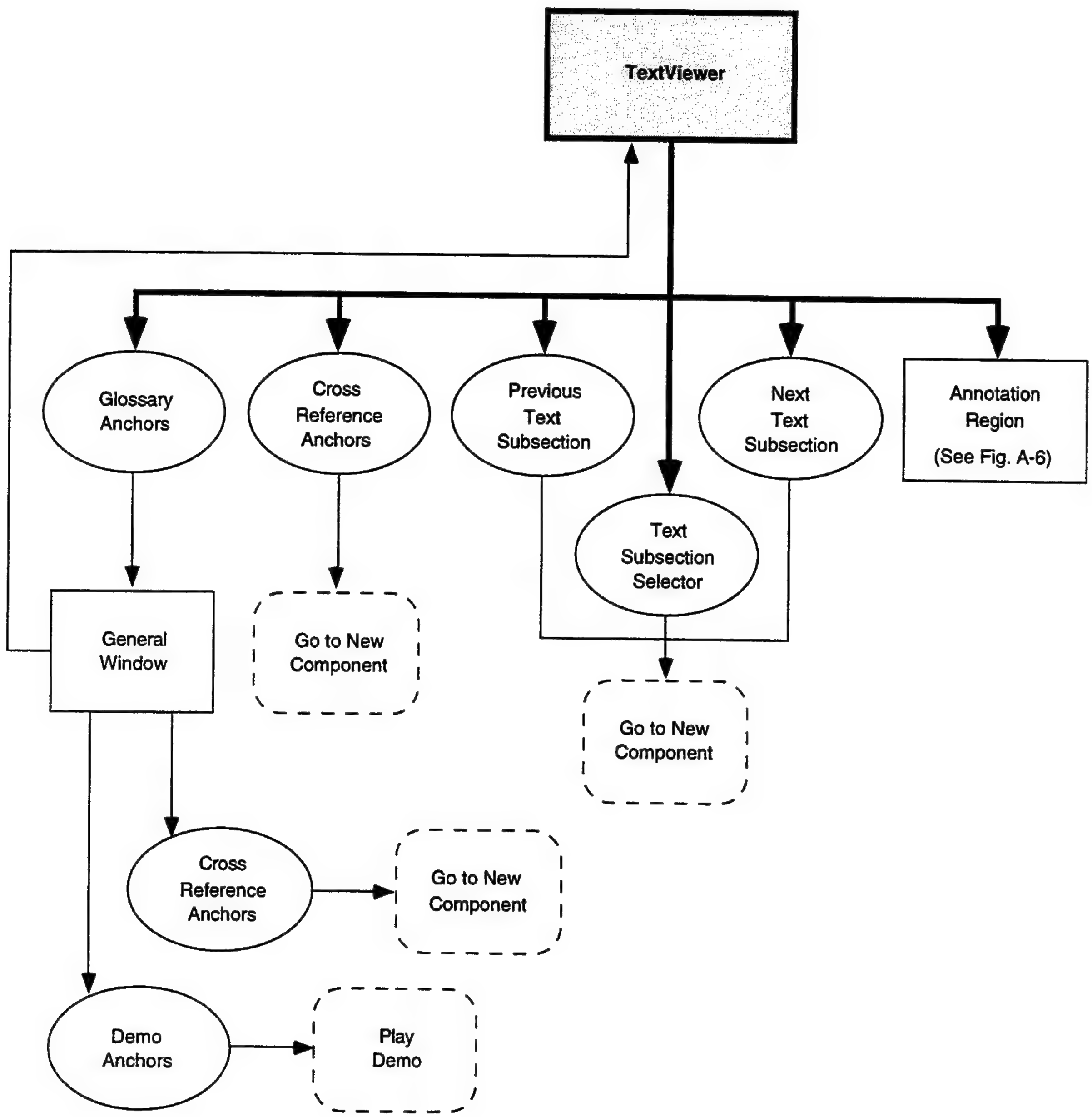


Figure A-1. Control flow for functions accessible from the TextViewer

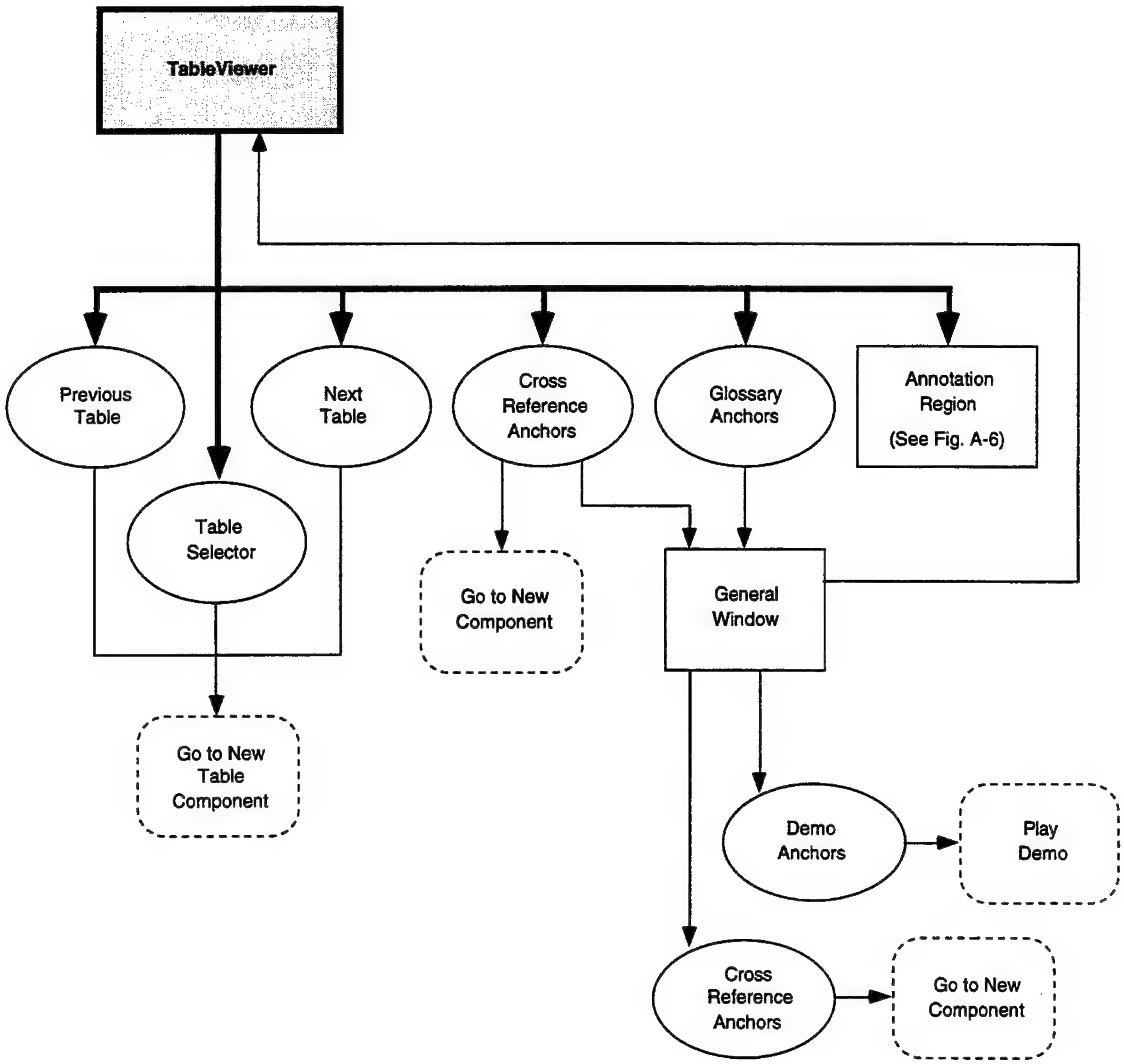


Figure A-2. Control flow for functions accessible from the TableViewer

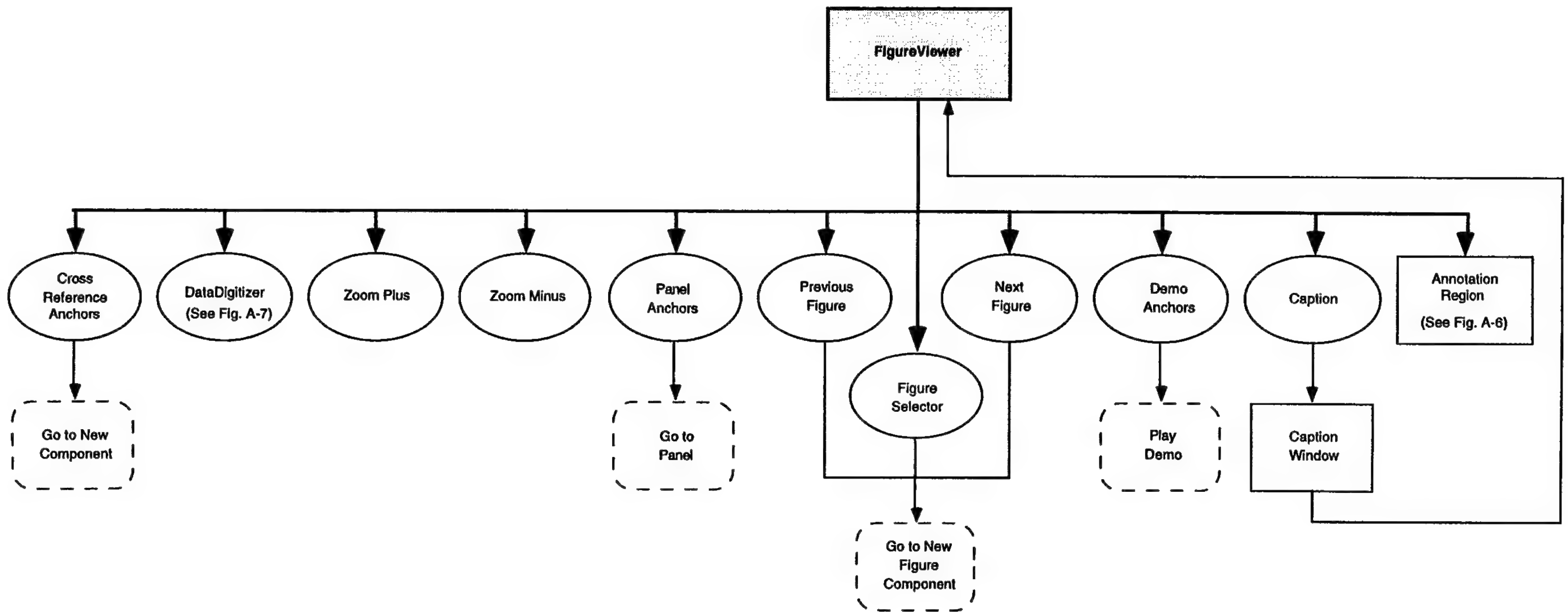


Figure A-3. Control flow for functions accessible from the FigureViewer

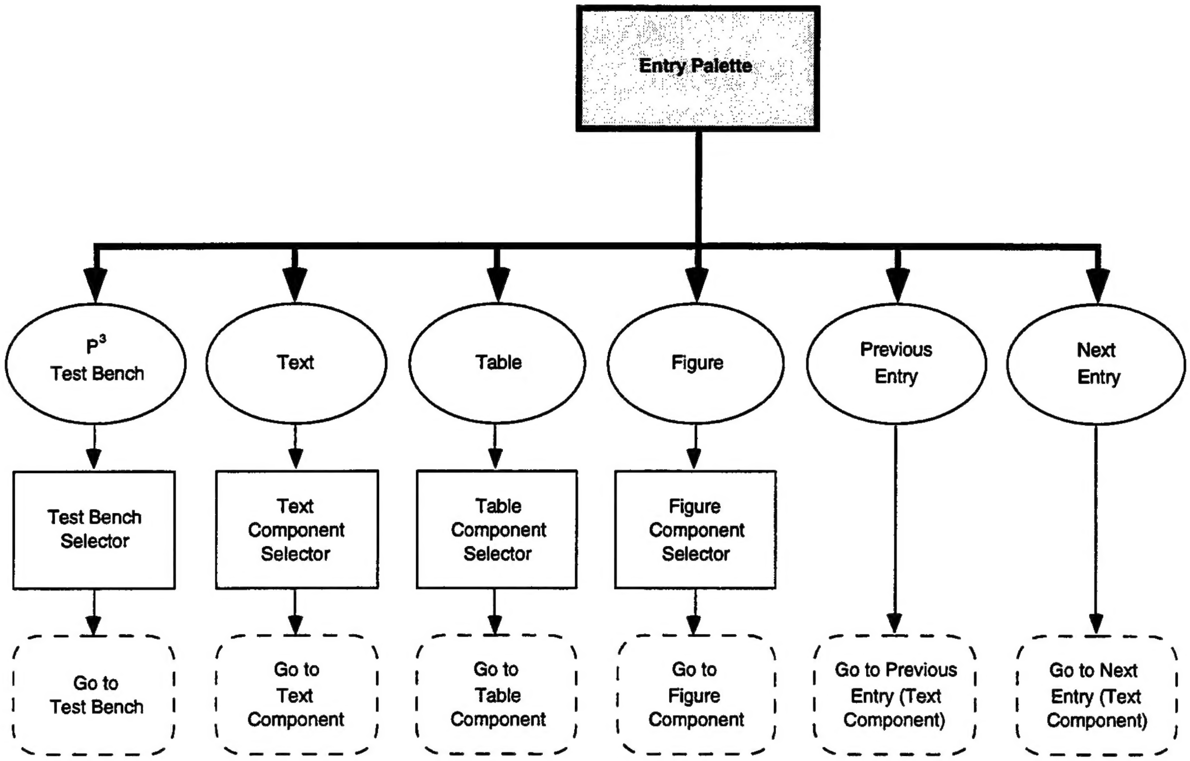


Figure A-4. Control flow for functions accessible from the entry palette

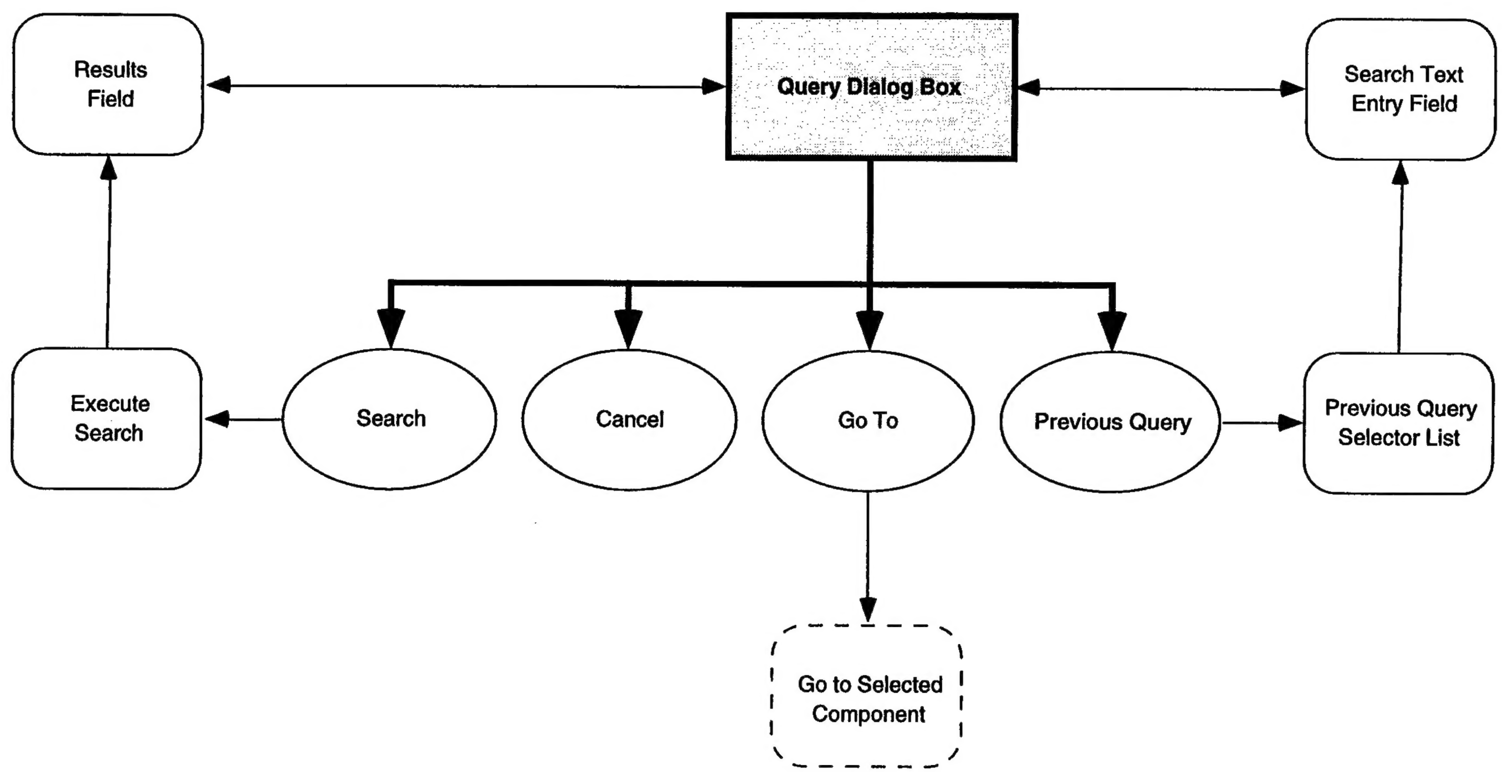


Figure A-5. Control flow for the Query function

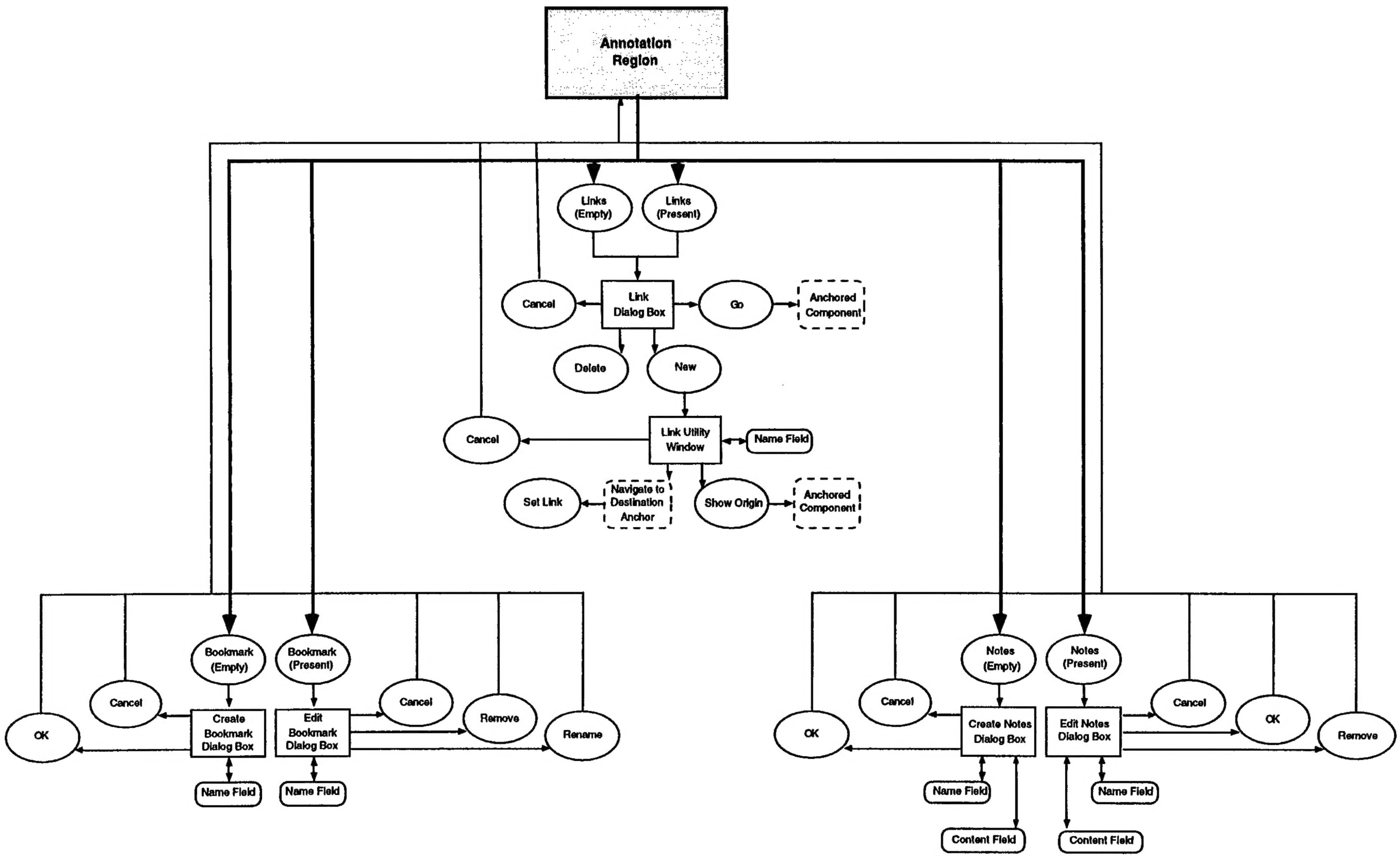


Figure A-6. Control flow for viewer annotation functions

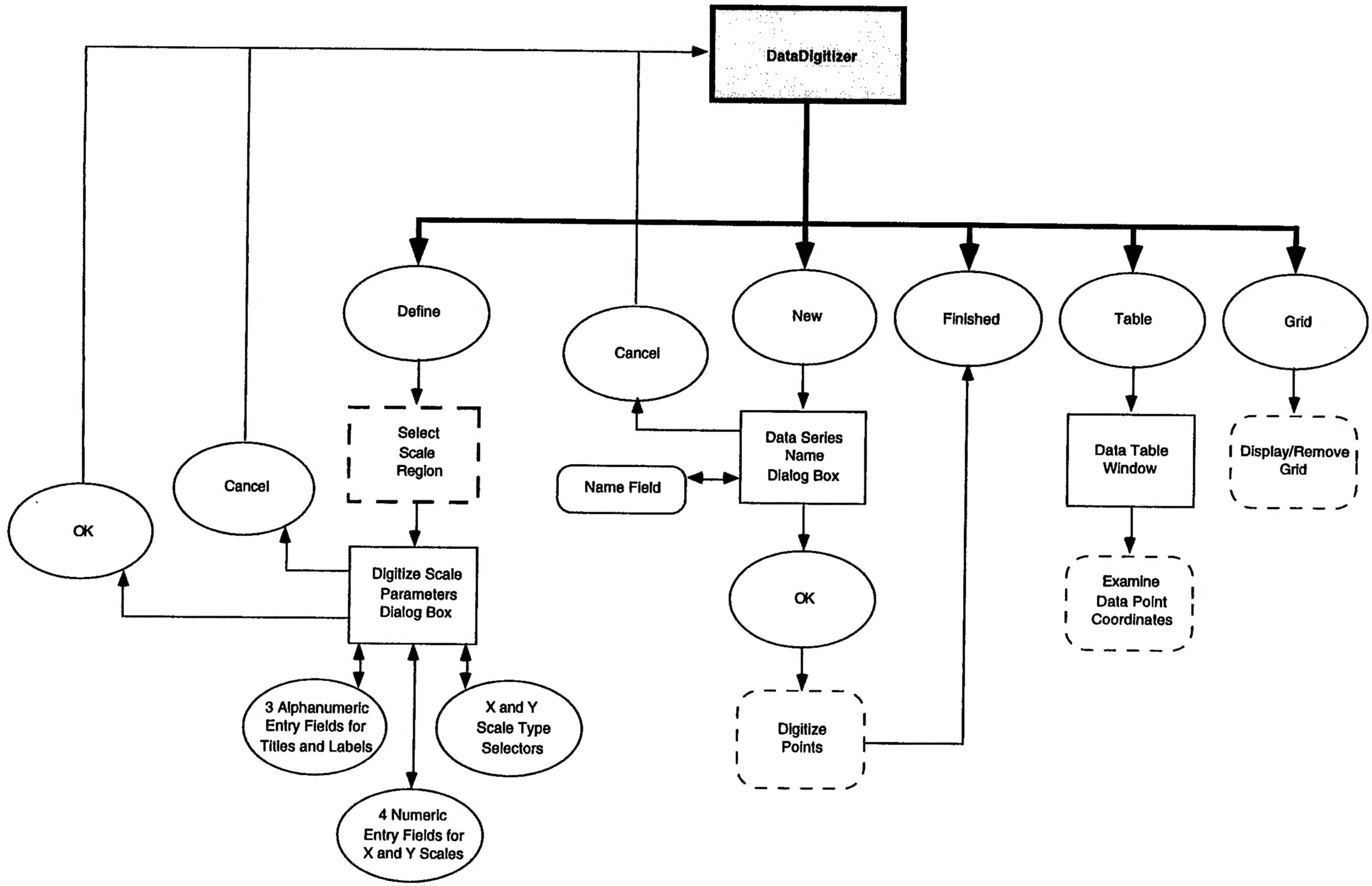


Figure A-7. Control flow for functions accessible from the DataDigitizer